

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри

О.А.Павлов

(підпис)

(ініціали, прізвище)

“ ”

2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки 6.050103 «Програмна інженерія»

на тему Комплекс програм для віддаленого доступу до бази знань

патернів аномалій роботи серця людини.

Виконав: студент IV курсу, групи Сорочинський Р.В.

(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц. Баклан І.В.

посада, науковий ступінь, вчене звання, прізвище, ініціали

(підпис)

**Консультант
з графічної
документації**

доц., к.т.н., Лішук К.І

посада, науковий ступінь, вчене звання, прізвище, ініціали

(підпис)

Рецензент:

доц. каф. ТК, к.т.н., доц. Ткач М.М.

(посада, науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2019 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – 6.050103
«Програмна інженерія» (Програмне забезпечення систем)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри

О.А. Павлов
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Сорочинському Руслану Володимировичу
(прізвище, ім'я, по батькові)

1. Тема проекту «Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини»

керівник проекту доц. Баклан І.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “16” квітня 2019 р. №1153-с

2. Термін подання студентом проекту «04» червня 2019 року

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: основні визначення та терміни, опис предметного середовища, огляд існуючих технічних рішень та відомих програмних продуктів, розробка функціональних та нефункціональних вимог

2) Моделювання та конструювання програмного забезпечення: моделювання та аналіз програмного забезпечення, засоби розробки, технічні рішення, архітектура програмного забезпечення

3) Розгортання та впровадження програмного забезпечення

4) Керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1) Схема структурна варіантів використань

2) Схема структурна станів системи

3) Схема структурна варіантів Використань з функціональними вимогами

4) Схема структурна послідовностей

5) Схема бази даних

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «28» лютого 2019 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури	28.02.19	
2.	Аналіз існуючих методів розв'язання задачі	07.03.19	
3.	Постановка та формалізація задачі	12.03.19	
4.	Аналіз вимог до програмного забезпечення	17.03.19	
5.	Алгоритмізація задачі	25.03.19	
6.	Моделювання програмного забезпечення	29.03.19	
7.	Обґрунтування використовуваних технічних засобів	04.04.19	
8.	Розробка архітектури програмного забезпечення	19.04.19	
9.	Розробка програмного забезпечення	02.05.19	
10.	Налагодження програми	06.05.19	
11.	Виконання графічних документів	09.05.19	
12.	Оформлення пояснювальної записки	22.05.19	
13.	Подання ДП на попередній захист	29.05.19	
14.	Подання ДП рецензенту	05.06.19	
15.	Подання ДП на основний захист	19.06.19	

Студент _____ Сорочинський Р.В.
(підпис)

Керівник проекту _____ Баклан І.В.
(підпис)

[illegible]

Пояснювальна записка до дипломного проекту

на тему: Комплекс програм для віддаленого доступу до бази знань патернів аномалій
роботи серця людини

Київ – 2019 року

АНОТАЦІЯ

До бакалаврської дипломної роботи Сорочинського Руслана Володимировича

На тему: «Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини»

Об'єкт дипломного проекту: захворювання серця у контексті розбору електрокардіограми.

Мета дипломного проекту: розробка веб-застосування для виявлення патологій серця на базі знімку електрокардіограми.

У першому розділі наведено аналіз предметної області, а саме:

- надано інформацію про лінгвістичне моделювання взагалі;
- проведено аналіз конкурентних проектів та описані їх деталі;
- розбір ЕКГ знімку і отримання ЕКГ-лексми із використанням комп'ютерного зору;

- аналіз вимог до програмного забезпечення.

У другому розділі наведено:

- використана архітектура;
- діаграма класів та їх опис;
- схему бази даних і опис таблиць;
- використані підходи до написання програми.

У третьому розділі наведений підхід до тестування: таблиця SmokeCheckList, тобто сценарії тестування, та таблиця знайдених помилок. Окрім цього, описано процес юніт-тестування.

У четвертому розділі представлено, яким чином розгортати дане програмне забезпечення та які мінімальні вимоги до програмних засобів для цього. Також наведена інструкція користувача і інструкція програміста.

Ключові слова: КОМП'ЮТЕРНИЙ ЗІР, ЕКГ, ЗНІМОК ЕЛЕКТРОКАРДІОГРАМИ, ЗАХВОРЮВАННЯ СЕРЦЯ, ПАТОЛОГІЯ

					КПІ-ІП-34111.045490.02.81	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

To the bachelor thesis of Sorochinsky Ruslan Volodimirovich.

On the topic: "A complex of programs for remote access to the knowledge base of patterns of abnormalities of human heart work "

The object of study: heart diseases in the context of parsing the electrocardiogram.

The aim of the degree project: development of the web application for the heart pathologies detection based on an electrocardiogram scan picture.

The first section provides domain analysis:

- information on linguistic modeling in general has been provided;
- analysis of competitive projects and their details have been described;
- analysis of ECG images and receiving ECG tokens using computer vision;
- software requirements analysis.

The second section contains:

- used architecture;
- a diagram of classes and their description;
- database schema and descriptions of tables;
- used approaches to writing the program.

The third section provides an approach to testing: the SmokeCheckList table, that consists of test scripts, and the table of issues found. In addition, the process of unit-testing is described.

The fourth section shows how to deploy this software and what are the minimum software requirements for this. Also a user manual and a programmer's manual are provided.

Keywords: COMPUTER VISION, ECG, ELECTRIC CARDIOGRAM SCAN, HEART DISEASES

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	6
ВСТУП	8
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	10
1.1 Загальні положення.....	10
1.2 Змістовний опис і аналіз предметної області.	12
1.3 Аналіз успішних ІТ-проектів.....	21
1.3.1 Аналіз відомих технічних рішень.....	21
1.3.2 Аналіз відомих програмних продуктів	22
1.3.3 Розроблення функціональних вимог.	35
1.3.4 Розроблення нефункціональних вимог.....	46
1.4 Висновки по розділу	47
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	48
2.1 Моделювання та аналіз програмного забезпечення.....	48
2.2 Архітектура програмного забезпечення	52
2.2.1 Платформа ASP.NET core	49
2.2.2 Структура MVC	53
2.2.3 Схема БД та опис таблиць.....	55
2.2.4 Migrations	66
2.2.3 Models	67
2.3 Аналіз безпеки даних.....	65
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	71
3.1 Перелік юніт-тестів для основного функціоналу	71

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	70
4.1 РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	70
4.1.1 Мінімальна конфігурація технічних засобів	70
4.1.2 Вимоги до операційних систем	70
4.1.3 Вимоги до інформаційної та програмної сумісності	70
4.1.4 Вимоги до маркування та пакування	70
4.1.5 Вимоги до транспортування та зберігання	70
4.1.6 Спеціальні вимоги.....	70
4.1.7 Публікація програмного забезпечення	71
4.2 РОБОТА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	73
ВИСНОВКИ.....	74
РОЗРОБКА ТЕХНІЧНОЇ ДОКУМЕНТАЦІЇ ЗА ДСТУ.ПЕРЕЛІК ПОСИЛАНЬ	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕКГ – електрокардіограма.

Скан ЕКГ – знімок електрокардіограми.

БД – база даних.

ПЗ – програмне забезпечення.

Технологічний стек – технології та інструменти, що були використані при розробці програмного продукту.

Email – електронна скринька.

Акаунт – обліковий запис (користувача).

Режим офлайн – режим за відсутності підключення до мережі.

Десктопний (англ. desktop - настільний) – вид ПЗ, що встановлюється на комп'ютер, а не виконується в браузері.

TDD (англ. Test Driven Development – розробка через тестування) – техніка розробки ПЗ, де потрібно спочатку написати тест, який не працює, перш ніж написати новий функціональний код, а згодом змінити тест так, щоб він працював і покривав новий функціонал.

CodeFirst – підхід до побудови БД, в якому спочатку описуються моделі, з яких за допомогою EntityFramework генеруються скрипти для бази даних.

Razor – двигун генерації html коду із C# коду, який використовується в ASP.NET фреймворку.

API – програмний інтерфейс.

CI (англ. continuous integration – безперервна інтеграція) - у процесі розробки програмного забезпечення практика злиття всіх робочих копій розробника на спільну гілку розробки кілька разів на день. Екстремальне програмування (XP) прийняло концепцію CI і виступає за інтеграцію неодноразово на день - можливо, десятки разів на день [7].

Деплой - задача розгортання програми на новій машині / або на тій же самій, але нової її версії.

Open-source software (OSS) - це тип комп'ютерного програмного забезпечення, вихідний код якого випущений за ліцензією, в якій власник авторських прав надає користувачам права вивчати, змінювати та розповсюджувати програмне забезпечення будь-кому та з будь-якою метою [9].

					КПІ-ІП-34111.045490-02-81	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

ВСТУП

На сьогодні, найбільш актуальним та розповсюдженим вважається ПЗ із графічним інтерфейсом у вигляді веб-застосунку. На відміну від ПЗ консольного типу, яке значно застаріло, веб-застосунки значно легші у використанні. В даній дипломній роботі було розроблено веб-застосунок для людей, що хочуть перевірити стан свого серця та виявити його потенційні слабкі сторони у режимі реального часу. Детальніше продукт описано у першому розділі.

Проаналізувавши медичну статистику по кардіології, можна побачити, що кількість людей із хворобами серцево-судинної системи щорічно неухильно зростає. Величезну роль у виявленні патологій серця на ранніх стадіях може зіграти аналіз даних електрокардіограми. Тому розробка програми, за допомогою якої будь-яка людина з легкістю дізнається про наявність хвороби у своєму організмі, завантаживши знімок електрокардіограми зі свого смартфона чи комп'ютера, зменшить черги до лікаря, а процес виставлення діагнозу та лікування стане значно швидшим.

Це стандартні задачі медицини та біомедичної інженерії. Їх було вирішено реалізувати у вигляді клієнт-серверного застосунку на платформі ASP.NET Core на стороні сервера та React частини на стороні клієнта (браузера). Більш детально про технологічний стек та використані інструменти описано у другому розділі.

Розробка призначена для медичних установ - лікарень, госпіталів, діагностичних центрів, пологових будинків, тощо, для таких функцій: оперативний аналіз знімків ЕКГ із подальшим збереженням результатів, збору та перегляду статистики.

Метою створення розробки є, в першу чергу, прискорення та автоматизація процесів розбору ЕКГ. Окрім цього, робота даного програмного продукту у перспективі спрямована також на автоматизацію

процесів ведення облікових записів користувачів із можливою монетизацією у вигляді деяких платних функцій, як то розширена статистика чи поради від робота-лікаря зі штучним інтелектом.

Як відомо, невід’ємною частиною процесу розборки є тестування програмного забезпечення. Як замовник, так і розробник, повинні бути впевнені, що програмний продукт відповідає вимогам, висуненим в технічному завданні, а найбільш доцільним способом переконатись у цьому є проведення ефективного тестування. Тому при розробці застосунку було виконано тестування за наявними вимогами, а опис надано у третьому розділі.

					КПІ-ІП-34111.045490-02-81	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дат		

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Перші цифрові електрокардіографи мали всього одну, але дуже важливу функцію - реєстрація, відображення та зберігання електрокардіограми в цифровій формі. Друге покоління забезпечувало вже можливістю автоматичного розпізнавання інформативних фрагментів сигналу і вимірювання традиційних електрокардіографічних ознак серцево-судинних захворювань. І, нарешті, з'явилися цифрові електрокардіографи, в яких були вбудовані алгоритми підтримки прийняття рішень лікаря-кардіолога, засновані на попередньому досвіді візуальної інтерпретації.

Існують різні підходи до комп'ютерного аналізу форми інформативних фрагментів ЕКГ. Найчастіше для такого аналізу використовуються алгоритми апроксимації сигналу, що дозволяє апроксимувати елементи ЕКГ поліноміальними функціями. Інші алгоритми надають можливість опису ЕКГ сплайн-функціями або за допомогою похідних. Решта використовує цифрові фільтри, або базуються на площах і дотичних.

Діагностика серцево-судинної системи людини належить до числа найважливіших завдань кардіології. Основні причини смертності людей працездатного віку пов'язані з серцево-судинними захворюваннями. Цим обумовлена необхідність розробки і вдосконалення засобів моніторингу для об'єктивного оцінювання і прогнозування стану серцево-судинної системи. На даний момент ЕКГ є найбільш розповсюдженим методом діагностики роботи серцево-судинної системи людини.

Центральний елемент системи кровообігу ссавців - серце - порожнистий м'язовий орган, здатний до ритмічних скорочень, забезпечує

безперервний рух крові всередині судин. Скорочення серця супроводжуються його електричною активністю. Останню можна реєструвати з поверхні тіла людини методом електрокардіографії. Дамо коротку характеристику інформативним фрагментами циклу ЕКГ.

Електрокардіографія — це метод графічної реєстрації електричних явищ, які виникають у серцевому м'язі під час його діяльності, з поверхні тіла. Криву, яка відображає електричну активність серця, називають електрокардіограмою (ЕКГ). Таким чином, ЕКГ — це запис коливань різниці потенціалів, які виникають у серці під час його збудження.

У літніх осіб має місце висока частота станів, що супроводжуються зміною рівня свідомості. Це пояснює протилежну тенденцію до гіпердіагностики епілепсії. Якщо в загальній групі хворих, направлених до епілептологічного кабінету, пацієнти з не епілептичними нападами склали 16,5%, то серед осіб похилого віку - 27,5%. Найчастіше за епілептичні напади приймалися кардіогенні синкопи (кардіогенна непритомність), обумовлені порушенням серцевого ритму. Їхня частка склала 7,8%. У пацієнтів із такими нападами був обтяжений кардіологічний анамнез (перенесений інфаркт міокарда, локалізований або дифузний кардіосклероз, серцева недостатність).

Відсутність таких симптомів означає, що серце людини здорове. Ідеалізована форма циклу ЕКГ здорової людини показана на рисунку 1.1.

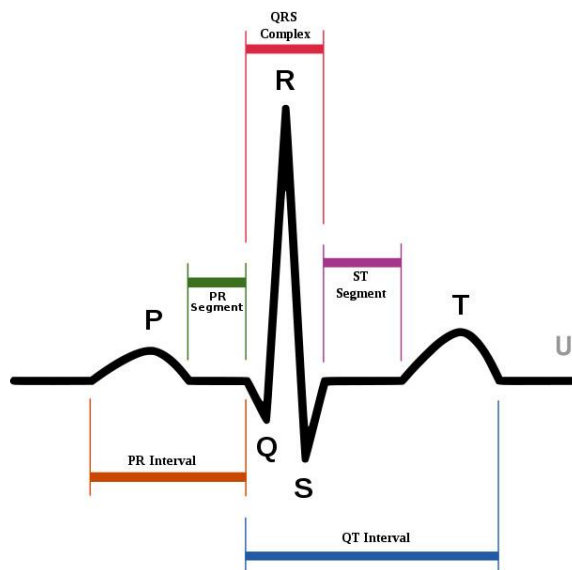


Рисунок 1.1 – Ідеалізована форма циклу ЕКГ здорової людини

1.2 Змістовний опис і аналіз предметної області.

Призначення системи аналізу кардіограм є:

- допомога людям виявляти аномалії у роботі серця;
- створення бази знань з типовими аномаліями.

Головні завдання, що вирішуються системою:

- створення бази лінгвістичних моделей порушень роботи серця;
- перетворення довільної кардіограми до вигляду лінгвістичного ланцюга;
- реалізація системи діагностики (виявлення лінгвістичних образів порушень роботи серця в кардіограмі у вигляді лінгвістичного ланцюга).

Для вирішення поставлених задач в системі були розроблені наступні підсистеми:

- підсистема обробки лінгвістичного словника;
- підсистема збереження бази даних.

В підсистемі обробки лінгвістичного словника реалізований функціонал за допомогою якого можна на основі вже наявних даних

будувати логістичну регресію тим самим визначаючи ймовірність того що завантажене ЕКГ містить аномалії.

В підсистемі збереження бази даних реалізоване збереження даних (ЕКГ, лінгвістичного словника, інформації про користувача).

Головним завданням лінгвістичного моделювання є перетворення чисельних рядів, експериментальних даних, багатомірних даних до лінгвістичних послідовностей та відновлення за ними формальної граматики мови відповідного характеру для вирішення наступного спектру проблем: аналіз та прогнозування часових рядів, розпізнавання образів різноманітної природи, автентифікація користувача за його рухами, розпізнавання емоційного стану оператора, діагностика хвороби опорно-рухового апарату операторів складних технічних систем на ранніх стадіях захворювання.

Лінгвістичне моделювання базується на трьох основних підходах: структурний підхід та математична лінгвістика, інтервальні обчислення та робастні методи, сучасні методи ймовірнісного моделювання.

В основі лінгвістичного моделювання лежить лема існування ізоморфізма відворення чисельних даних до лінгвістичних послідовностей, на основі яких може бути побудована мова. Як висновок існування унікальної мови, яка фактично уособлюється наборами чисельних даних.

Лінгвістична модель — побудована на основі лінгвістичного моделювання сукупність символічних (лінгвістичних) послідовностей за обраними параметрами лінгвістизації та відновлена на її основі формальна граMATика.

Лінгвістичне моделювання треба розглядати як специфічний вид математичного моделювання для обробки даних у символічному (не чисельному) вигляді.

Головна ідея переходу від множини чисельних значень до певного символічного алфавіту є розбиття цієї множини на чисельні інтервали.

Нехай X та Y - дві частково впорядкованих множини. Позначимо через

$B(X)$ та $B(Y)$ множини усіх підмножин множин X та Y .

Множина, на якій задано відношення порядку, має назву впорядкованої, якщо відношення порядку визначено для будь-яких двох його елементів, а частково впорядкованою у протилежному випадку.

Частково впорядкована множина називається структурою, якщо будь-яка її двоелементна підмножина має точну верхню та нижню межу, а повною структурою, якщо кожна її непушта підмножина має такі точні межі.

Кожну з множин X та Y будемо вважати умовно повними структурами та позначати $S(X)$ та $S(Y)$. Відношення порядку будемо позначати через \leq .

Якщо $a, b \in S(X)$ та $a \leq b$, то множину $l(a, b) = [a, b] = \{x \in X, a \leq x \leq b\}$, будемо називати інтервалом на $S(X)$.

Множину усіх інтервалів на структурі $S(X)$ будемо позначати $\zeta_{S(X)}$. При цьому, якщо $X = R^1$ - множина дійсних чисел, то ζ_{R^1} - множина закритих інтервалів на прямій дійсних чисел. В такому випадку називають інтервальних числом.

Для даної системи інтервали мають бути рівночастотні. Тобто при рівнозначній інтевралізації N -того рівня множини X :

$$\omega(a_1, b_1) = \omega(a_2, b_2) = \omega(a_3, b_4) = \dots = \omega(a_N, b_N) \quad 1.1$$

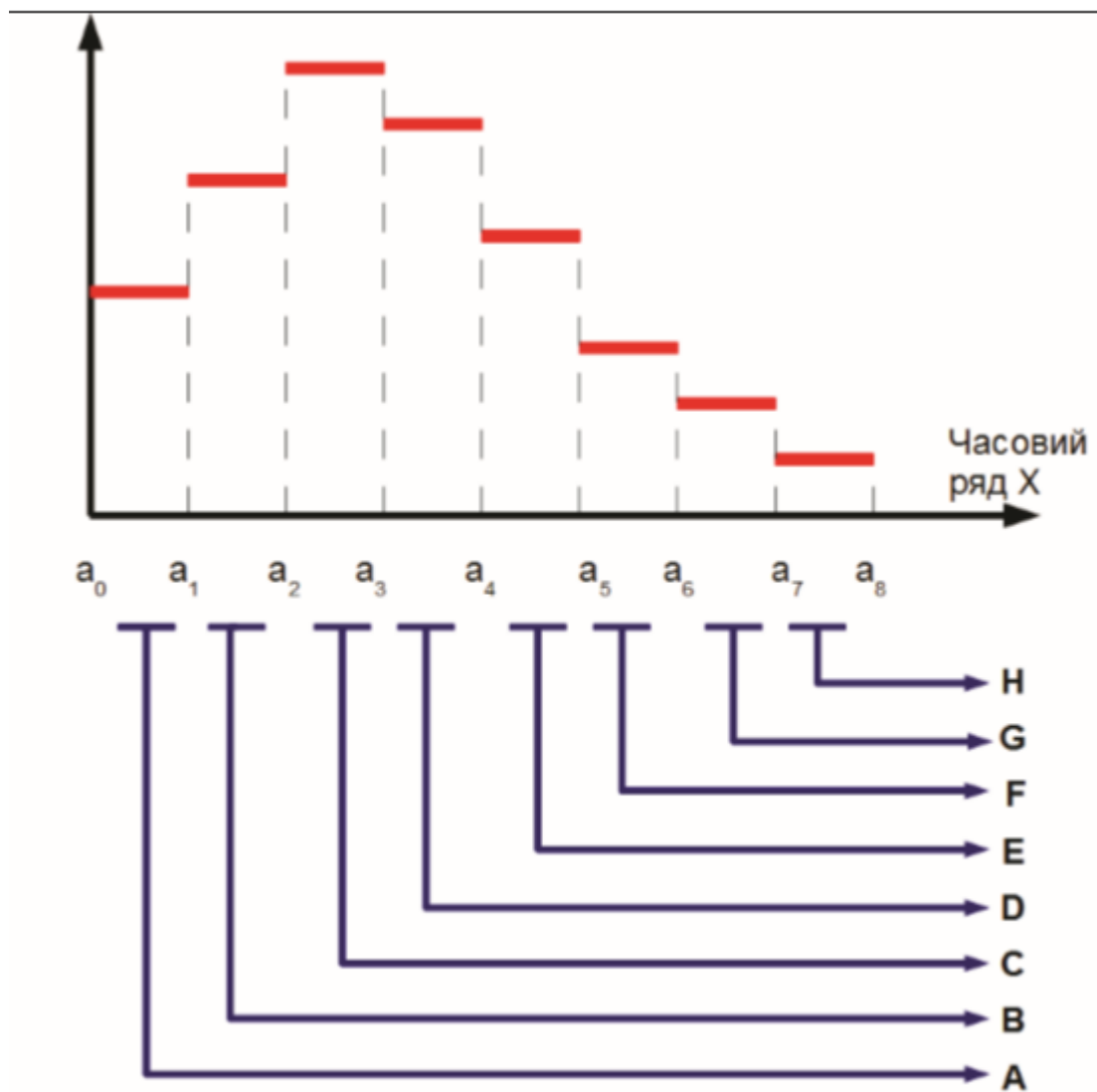


Рисунок 1.2 - Схема перетворення чисельних даних до символічного вигляду

На рисунку 1.2 представлена умовна схема лінгвістизації при рівнозначній інтервалізації. Інтервалізація дає множину інтервалів $I[a_0, a_1], I[a_1, a_2], \dots, I[a_7, a_8]$. За віссю абсцис розташовані впорядковані значення часового ряду та інтервали. За віссю ординат ранжується кількість значень часового ряду, що потрапили до відповідного інтервалу. Кількість елементів, які потрапляють до певного інтервалу (значення часового ряду можуть повторюватися) будемо позначати через $D\{I[a_i, a_{i+1}]\}$. Кількість інтервалів - $M < N$. Легко бачити, що сума $\sum_{i=0}^{M-1} D\{I[a_i, a_{i+1}]\} = N$

Тепер ми можемо ввести поняття частотності інтервалу $I[a_i, a_{i+1}]$ на часовому ряді потужністю N :

$$v_{i,i+1} = v[a_i, a_{i+1}] = \frac{D\{I[a_i, a_{i+1}]\}}{N} \quad 1.2$$

Очевидно що, $\sum_{i=0}^{M-1} v_{i,i+1} = 1$, а враховуючи що для усіх i $v_{i,i+1} \geq 0$ будемо мати аналогію аксіоматики теорії ймовірностей.

Перейдемо до ЕКГ. Для дослідження довільної кардіограми потрібно виконати наступні кроки:

- дискретизація;
- векторизація;
- статистичне оцінювання.

Розберемо детальніше кроки дослідження:

Дискретизація:

Як було описано вище при аналізі предметної області було сказано стрічка рухається рівномірно, а на кожному інтервалі буде поставлена у відповідність числове значення зубців. Тобто, можна буде на кожному інтервалі часу поставити у відповідність область яку перетинає графік ЕКГ.

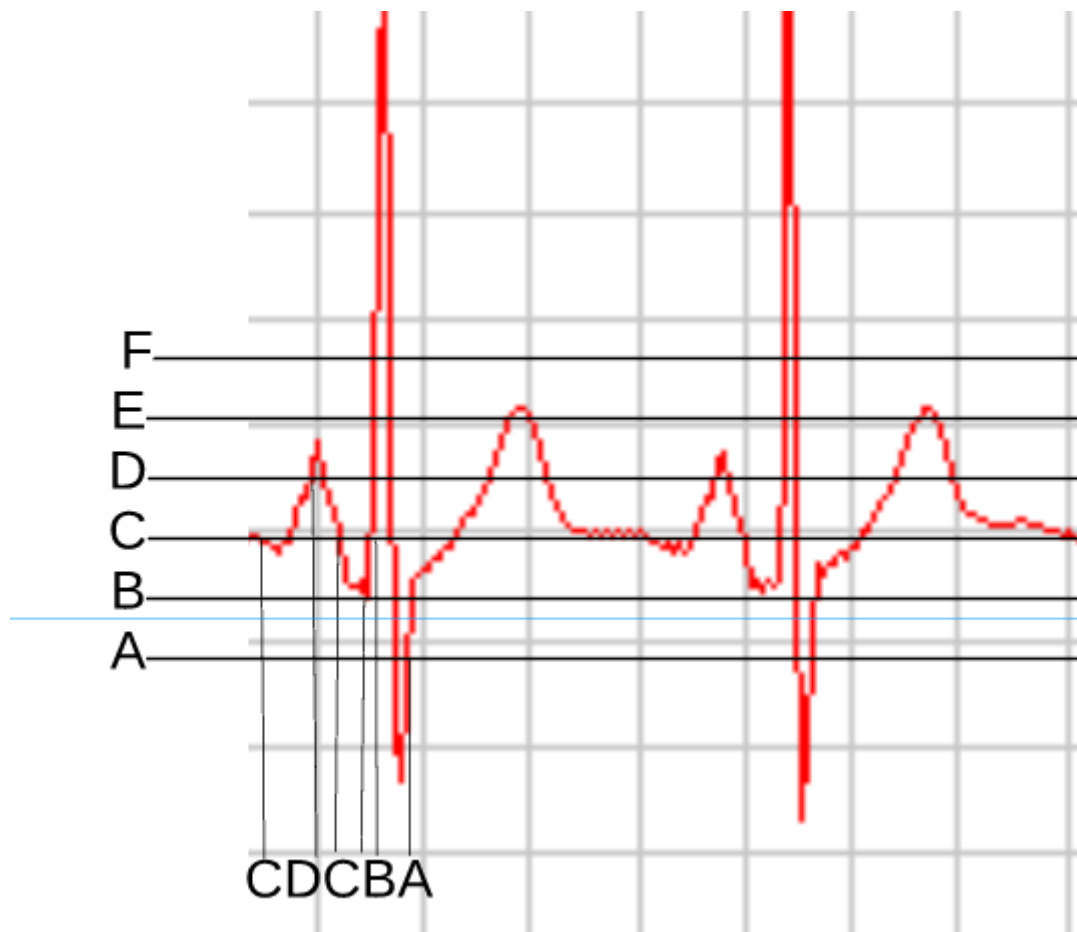


Рисунок 1.3 - Схематичне зображення перетворення ЕКГ до лінгвістичного вигляду

На рисунку 1.3 схематично показано фрагмент перетворення кардіограми у лінгвістичний вигляд. На виході отримаємо словник вигляду:

ABCCCGFDBC GGFDA...

Тобто на виході отримали кодограму $x = (s_n)_{n=1}^{N-1}$ – послідовність символів алфавіта $A = \{A, B, C, D, E, F\}$.

Векторизація

Даний етап полягає у створенні вектора триграм з кодограми отриманої на стадії дискретизації.

На вході маємо кодограму $x = s_1, \dots, s_{N-1}$ як текстову строку.

На виході отримаємо частоти триграм $f_j(x)$ – скільки разів триграма j появилася в кодограмі x , $j = 1, \dots, n$, $n = 6^3 = 216$

Наведем пример:

[illegible]

Рисунок 1.4 - Довільна кодограма

1. FFA - 42	17. EFF - 10	33. CEC - 6	49. EAC - 3
2. FAA - 33	18. DAA - 10	34. ADB - 5	50. DDA - 3
3. AFF - 32	19. ECF - 9	35. FFE - 5	51. CAC - 3
4. AAF - 30	20. FFC - 9	36. EBF - 5	52. EDF - 3
5. ADF - 18	21. FEA - 9	37. CFD - 5	53. EFB - 3
6. FCA - 18	22. DFC - 8	38. AFB - 4	54. DBA - 3
7. ACF - 17	23. ABF - 8	39. AAE - 4	55. FCC - 2
8. AAD - 15	24. AAB - 8	40. CFC - 4	56. AFC - 2
9. CFF - 14	25. FCE - 8	41. CAE - 4	57. EAA - 2
10. AEF - 13	26. AEB - 7	42. DAC - 4	58. CED - 2
11. FDA - 13	27. DFD - 7	43. DBF - 4	59. CAA - 2
12. FAE - 12	28. ACD - 6	44. BFC - 4	60. BCA - 2
13. FAC - 12	29. CDF - 6	45. CFB - 4	61. BBA - 2
14. FBA - 11	30. DFA - 6	46. AED - 3	62. DFF - 2
15. BFA - 11	31. CAF - 6	47. FFF - 3	63. BDA - 2
16. BAA - 11	32. CAD - 6	48. FBC - 3	64. DAE - 2

Рисунок 1.5 - Частота триграм

На рисунку 1.4 зображення довільна кодограма, а на рисунку 1.5 частоти триграм для неї.

Статистичне оцінювання

На попередніх кроках було отримано кодограму і відповідно до неї триграму. Для подальшого дослідження потрібно щоб у базі були кодограми для еталонного значення наприклад 100 кодограм здорової людини і 100 хворої.

Далі можна проводити подальший аналіз наприклад побудови логістичної регресії.

Вона дозволяє оцінювати належність об'єкту до певного класу.

На Рисунку 1.6 наведена схема структурна діаграми активності процесу аутентифікації.

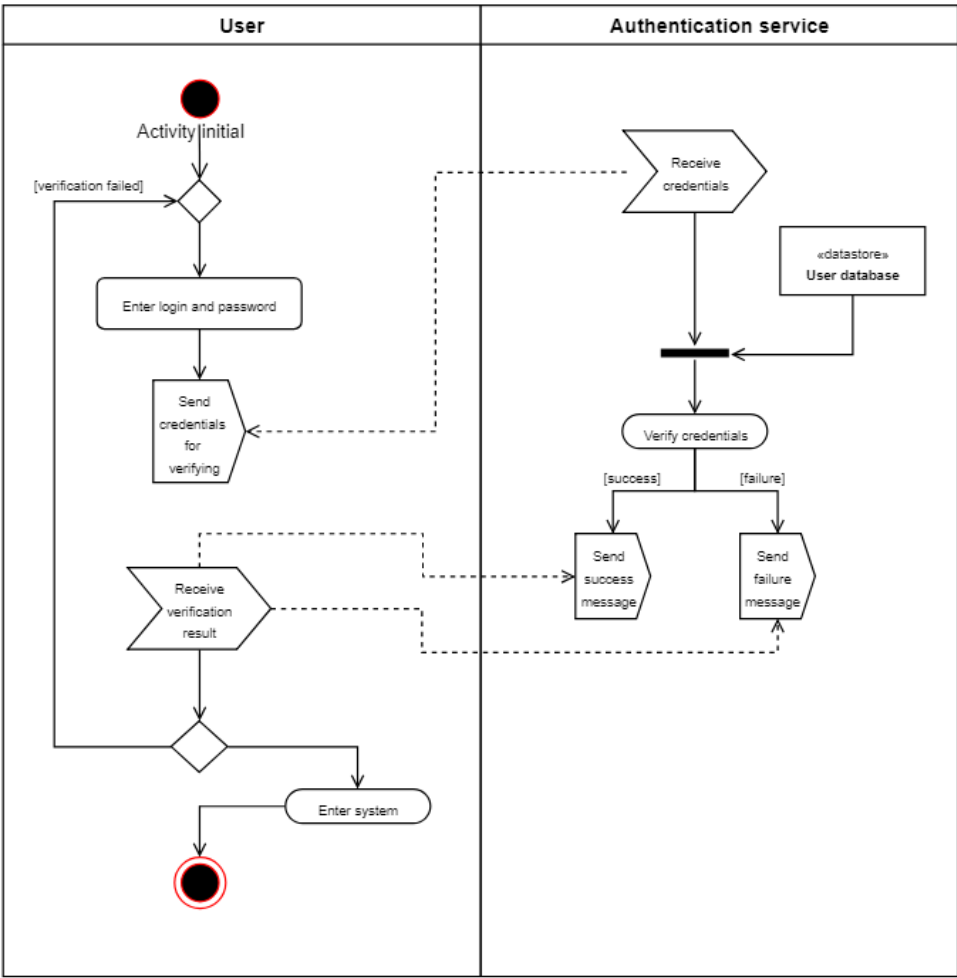


Рисунок 1.6 - Схема структурна активності процесу аутентифікації

1.3 Аналіз успішних ІТ-проектів

1.3.1 Аналіз відомих технічних рішень

Загалом для вирішення поставлених задач використовується наступний підхід: створюються API-кінцеві точки (endpoints), що взаємодіють з даними через EntityFramework, які повертають дані у форматі JSON. На frontend за допомогою AJAX запитів цей JSON обробляється, і вже за допомогою JavaScript виводяться необхідні таблиці, графіки, елементи інтерфейсу.

1.3.2 Аналіз відомих програмних продуктів

Під час аналізу вимог даного дипломного проекту було проведено детальний пошук аналогічних сервісів у мережі Інтернет. Серед онлайн сервісів такого напрямку не було знайдено жодного. Встановлювані платні бізнес-рішення перераховані у таблиці 1.1

Таблиця 1.1 - Програми для аналізу та інтерпретації сигналу ЕКГ

№	Назва програми	Виробник	Ключові можливості
1	ECG interpretation software C	Schiller (Швейцарія)	Більше 100 різних варіантів інтерпретації ЕКГ
2	Signal-Averaged ECG Software (SAECG)	Schiller (Швейцарія)	Аналіз сигнал-усередненої ЕКГ
3	Heart Rate Variability (HRV) Software	Schiller (Швейцарія)	Аналіз варіабельності серцевого ритму (ЕКГ)
4	Marquette 12SL	GE Healthcare (США)	Аналіз ЕКГ в 12 відведеннях; унікальні критерії оцінки ST- і QT-сегментів і зубця Т; автоматичне виявлення аритмій
5	Cardiosoft	GE Healthcare (США)	Розширений аналіз сегмента ST; аналіз альтернанса зубця Т; автоматичне і ручне вимір інтервалів ЕКГ; аналіз аритмій по 2 відведенням; унікальні алгоритми фільтрації шумів і вирівнювання ізолінії

Продовження таблиці 1.1

6	FP-804	Fukuda (Японія)	Основні вимірювання: ЧСС, RR, PR, QRS, час QT, QTc, електрична вісь, SV, RV5 (6); введення інформації про пацієнта; 120 типів кодів інтерпретації і 130 типів кодів Міннесоти
---	--------	--------------------	---

Через відсутність конкурентних додатків остаточно було вирішено розробити саме онлайн платформу для обробки сканів ЕКГ у режимі реального часу.

Аналіз вимог до програмного забезпечення

Схема структурна варіантів використання наведена на Рисунку 1.7.

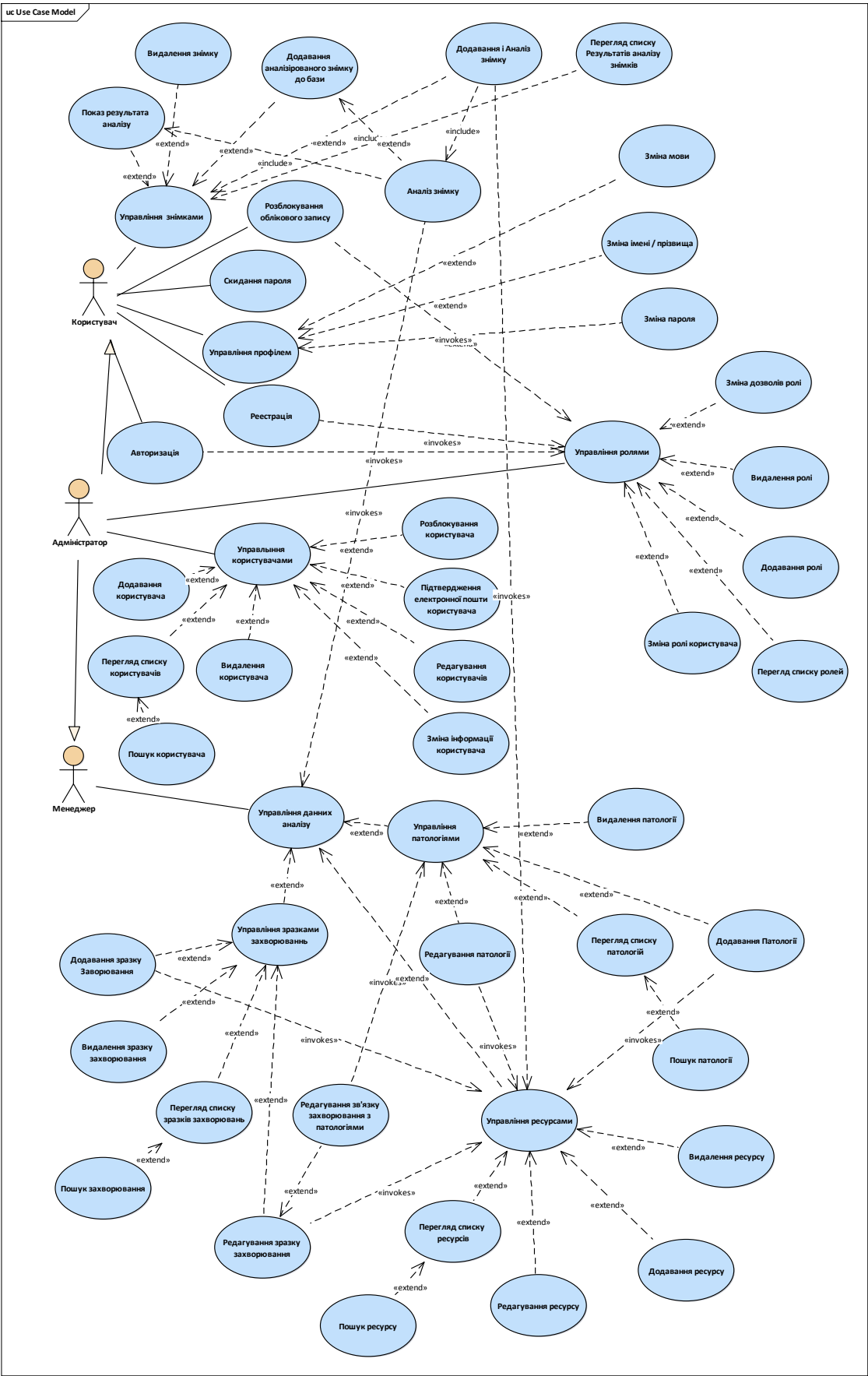


Рисунок 1.7 - Схема структурна варіантів використання

Опис варіантів використання.

- Ідентифікатор: 1.

Найменування: 1. Реєстрація.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Реєстрація.

Попередні умови: Відкрита сторінка реєстрації.

Вихідні умови: Користувачу на вказану при реєстрації пошту надіслано лист-підтвердження реєстрації.

- Ідентифікатор: 2.

Найменування: 2. Авторизація.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Аутентифікація.

Попередні умови: Відкрита сторінка аутентифікації, користувач зареєстрований та має підтверджену електронну пошту.

Вихідні умови: Користувач пройшов аутентифікацію та переправлений у власний кабінет, або йому видано повідомлення з відповідною помилкою.

- Ідентифікатор: 3.

Найменування: 3. Скидання пароля.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Відновлення паролю.

Попередні умови: Користувач зареєстрований, відображена сторінка відновлення паролю, користувач перейшов по посиланню у листі, що прийшов йому на пошту.

Вихідні умови: Пароль змінено та відновлено.

					КПІ-ІП-34111.045490-02-81	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дат		

- Ідентифікатор: 4.

Найменування: 4. Розблокування облікового запису.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Розблокування аккаунту.

Попередні умови: Користувач зареєстрований та підтверджений, користувач ввів більше 20 неправильних паролів, тобто користувач заблокований.

Вихідні умови: Якщо користувач перейде за посиланням, що було надіслано на його пошту, вказану при реєстрації, то він буде розблокований.

- Ідентифікатор 5.

Найменування: 5. Управління профілем.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Можливість змінювати власну інформацію.

Попередні умови: Користувач пройшов аутентифікацію, відкрита сторінка редагування інформації.

Вихідні умови: Інформацію змінено.

- Ідентифікатор: 5.1.

Найменування: 5.1. Зміна пароля.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Можливість змінити пароль.

Попередні умови: Користувач пройшов аутентифікацію, відкрита сторінка редагування інформації.

Вихідні умови: Пароль змінено.

- Ідентифікатор: 5.2.

Найменування: 5.2. Зміна імені / прізвища.

					КПІ-ІП-34111.045490-02-81	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дат		

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Можливість змінити ім'я та прізвище.

Попередні умови: Користувач пройшов аутентифікацію, відкрита сторінка редагування інформації.

Вихідні умови: Ім'я та прізвище змінено.

- Ідентифікатор: 6.1

Найменування: 6.1 Додавання і Аналіз знімку.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Форма Додавання знімку

Попередні умови: Користувач пройшов аутентифікацію та відображена форма додавання знімку.

Вихідні умови: Знімок подається на аналіз.

- Ідентифікатор: 6.2.

Найменування: 6.2. Аналіз знімку.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Аналізувати знімок.

Попередні умови: Користувач пройшов аутентифікацію та відображена форма додавання знімку.

Вихідні умови: Знімок аналізовано і трансформовано в результат аналізу знімка .

- Ідентифікатор: 6.3.

Найменування: 6.3. Показ результату аналізу.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Показує результат аналізу знімка.

Попередні умови: Користувач пройшов аутентифікацію та відображена форма додавання знімку і знімок аналізований та трансформований в результат аналізу знімка.

Вихідні умови: Знімок аналізовано і трансформовано в результат аналізу знімка .

- Ідентифікатор: 6.4.

Найменування: 6.4. Показ результату аналізу.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: результат аналізу знімка зберігається до бази даних (відбуваються паралельно 6.3).

Попередні умови: Користувач пройшов аутентифікацію та відображена форма додавання знімку і знімок аналізований та трансформований в результат аналізу знімка.

- Ідентифікатор: 6.5.

Найменування: 6.5. Перегляд списку Результатів аналізу знімків.

Первинний актор: Користувач.

Інші актори: Адміністратор.

Опис: Показує Користувачу Список Результатів Аналізу знімків.

Попередні умови: Користувач пройшов аутентифікацію та відображений особистий кабінет.

Вихідні умови: Відображено сторінку списку результатів аналізу власних попередніх знімків користувача.

- Ідентифікатор: 7.1.

Найменування: 7.1. Додавання патології.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Додає патологію до бази даних.

Попередні умови: Користувач пройшов аутентифікацію та відображена форма додавання патології.

Вихідні умови: Патологія зберігаються до бази та Відображено сторінку списку патологій.

- Ідентифікатор: 7.2.

Найменування: 7.2. Видалення патології.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Видаляє патологію з бази даних.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку списку патологій.

Вихідні умови: вибрана Патологія видаляється з бази та Відображено сторінку списку патологій.

- Ідентифікатор: 7.3.

Найменування: 7.3. Перегляд списку патологій.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Відображено сторінку списку патологій.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку списку патологій.

Вихідні умови: вибрана Патологія видаляється з бази та Відображено сторінку списку патологій.

- Ідентифікатор: 7.4.

Найменування: 7.4. Пошук патології.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Відображено фільтр пошуку патологій на сторінці списку патологій.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку списку патологій.

Вихідні умови: відбувається пошук патологій відповідно до критеріїв фільтра.

- Ідентифікатор: 8.1

Найменування: 8.1. Перегляд списку зразків захворювань.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Відображено сторінку списку зразків захворювань.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку особистого кабінету.

Вихідні умови: Відображено фільтр пошуку сторінці списку зразків захворювань.

- Ідентифікатор: 8.2

Найменування: 8.2. Пошук патології.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Відображено фільтр пошуку патологій на сторінці списку патологій.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку списку зразків захворювань.

Вихідні умови: відбувається пошук зразків захворювань відповідно до критеріїв фільтра.

- Ідентифікатор: 8.3

Найменування: 8.3 Видалення зразку захворювання.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Видаляє зразку захворювання з бази даних.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку списку зразків захворювань.

Вихідні умови: вибраний зразок видаляється з бази та Відображено сторінку списку зразків захворювань.

- Ідентифікатор: 8.4

Найменування: 8.4 Додавання зразку Захворювання.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Додає зразок Захворювання до бази даних.

Попередні умови: Користувач пройшов аутентифікацію та відображена форма додавання зразку Захворювання.

Вихідні умови: зразок Захворювання зберігаються до бази та Відображено сторінку списку зразків Захворювання.

- Ідентифікатор: 8.5

Найменування: 8.5 Редагування зразку захворювання.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Редагує зразок Захворювання і зберігає до бази даних.

Попередні умови: Користувач пройшов аутентифікацію та відображена форма Редагування зразку захворювання.

Вихідні умови: зразок Захворювання зберігаються до бази та Відображено сторінку списку зразків Захворювання.

- Ідентифікатор: 8.6

Найменування: 8.6 Редагування зв'язку захворювання з патологіями.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Редагує зв'язок захворювання з патологією і зберігає до бази даних.

Попередні умови: Користувач пройшов аутентифікацію та відображена форма Редагування зв'язку захворювання з патологіями.

Вихідні умови: зв'язок захворювання з патологією до бази та Відображено сторінку списку зразків захворювання.

- Ідентифікатор: 9.1

Найменування: 9.1. Перегляд списку ресурсів.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Відображено сторінку Перегляду списку ресурсів.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку особистого кабінету.

Вихідні умови: Відображено Перегляд списку ресурсів для подальших дій з ними.

- Ідентифікатор: 9.2

Найменування: 9.2. Пошук ресурсу.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Відображено фільтр пошуку патологій на сторінці Перегляд списку ресурсів.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку Перегляду списку ресурсів.

Вихідні умови: відбувається пошук ресурсів відповідно до критеріїв фільтра.

- Ідентифікатор: 9.3

Найменування: 9.3 Редагування зразку захворювання.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Редагує ресурс і зберігає до бази даних.

Попередні умови: Користувач пройшов аутентифікацію та відображена форма Редагування ресурсу.

Вихідні умови: ресурс зберігається до бази та Відображено сторінку списку ресурсів.

- Ідентифікатор: 9.4

Найменування: 9.4 Додавання ресурсу.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: Додає ресурс до бази даних.

Попередні умови: Користувач пройшов аутентифікацію та відображена форма додавання ресурсу.

Вихідні умови: ресурс зберігаються до бази та Відображено сторінку списку ресурсів.

- Ідентифікатор: 9.5

Найменування: 9.5 Видалення ресурсу.

Первинний актор: Менеджер.

Інші актори: Адміністратор.

Опис: видаляє ресурс з бази даних.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку списку ресурсів.

Вихідні умови: вибраний ресурс видаляється з бази та Відображено сторінку списку ресурсів.

- Ідентифікатор: 10.1

Найменування: 10.1. Перегляд списку Користувачів

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Відображено сторінку Перегляду списку Користувачів.

Попередні умови: Користувач пройшов аутентифікацію та Відображено сторінку особистого кабінету.

					КПІ-ІП-34111.045490-02-81	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дат		

Вихідні умови: Відображено Перегляд списку Користувачів для подальших дій з ними.

- Ідентифікатор: 10.2.

Найменування: 10.2. Створення користувача.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Створення користувача.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка створення користувача.

Вихідні умови: Користувач створений та йому на пошту висланий лист-підтвердження.

- Ідентифікатор: 10.3.

Найменування: 10.3. Видалення користувача.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Видалення користувача.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка перегляду користувачів.

Вихідні умови: Користувач видалений.

- Ідентифікатор: 10.4.

Найменування: 10.4. Редагування користувача.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Можливість редагувати користувача.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка редагування конкретного користувача.

Вихідні умови: Інформація користувача відредагована.

- Ідентифікатор: 10.4.1.

Найменування: 10.4.1. Зміна інформації користувача.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Можливість змінити інформацію користувача.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка редагування конкретного користувача.

Вихідні умови: Інформацію змінено.

- Ідентифікатор: 10.4.2.

Найменування: 10.4.2. Зміна ролі користувача.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Зміна ролей, до яких належить користувач(права доступу).

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка редагування конкретного користувача.

Вихідні умови: Ролі користувача змінено.

- Ідентифікатор: 10.4.3.

Найменування: 10.4.3. Підтвердження електронної пошти користувача.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Підтвердження пошти користувача, якщо вона не підтверджена.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка редагування конкретного користувача.

Вихідні умови: Підтверджена пошта користувача, якщо вона не підтверджена.

- Ідентифікатор: 10.5.

Найменування: 10.5. Розблокування користувача.

Первинний актор: Адміністратор.

Інші актори: немає.

					КПІ-ІП-34111.045490-02-81	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дат		

Опис: Розблокування заблокованого користувача.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка редагування конкретного користувача.

Вихідні умови: Користувач розблокований.

- Ідентифікатор: 10.6.

Найменування: 10.6. Пошук користувача.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Пошук користувача за критеріями фільтра.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка перегляду списку користувачів, поле пошуку.

Вихідні умови: користувач/користувачі знайдені.

- Ідентифікатор: 11.1

Найменування: 11.1 Перегляд списку ролей.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Перегляд списку ролей.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка перегляду ролей.

Вихідні умови: Ролі відображаються.

- Ідентифікатор: 11.2.

Найменування: 11.2. Додавання ролі.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Створення ролі.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка перегляду ролей.

Вихідні умови: Роль створена.

- Ідентифікатор: 11.3.

Найменування: 11.3. Видалення ролі.

Первинний актор: Адміністратор.

Інші актори: немає.

Опис: Видалення ролі.

Попередні умови: Адміністратор пройшов аутентифікацію та відкрита сторінка перегляду ролей.

Вихідні умови: Видалено роль.

- Ідентифікатор: 11.4.

Найменування: 11.4. Зміна дозволів ролі.

Первинний актор: Адміністратор.

Інші актори: немає..

Опис: Зміна прав доступу ролі.

Попередні умови: Користувач пройшов аутентифікацію та відкрита сторінка перегляду ролей.

Вихідні умови: Права доступу змінені.

1.3.3 Розроблення функціональних вимог.

Таблиця 1.2 – Функціональні вимоги для актора Користувач

Варіант використання	Функціональні вимоги
1. Реєстрація	RQ 001.1. Система надає користувачу форму для створення аканту з такими полями: електронна пошта, пароль, підтвердження паролю, ім'я, прізвище.
	RQ 001.1.1. Система має перевіряти пароль і підтвердження паролю, щоб вони співпадали і надавати користувачу повідомлення якщо це не так.

Продовження таблиці 1.2

	RQ 001.1.2. Система має перевіряти, щоб введена електронна пошта була у форматі електронної пошти і надавати користувачу повідомлення якщо це не так.
	RQ 001.1.3. Система має перевіряти чи користувач з таким mail існує і надавати користувачу повідомлення
	RQ 001.1.4. Система має перевіряти чи введений пароль більше 8 символів.
	RQ 001.1.5. Система має перевіряти чи email, пароль і підтвердження паролю введені, бо ці поля обов'язкові.
	RQ 001.2.1. При переході за посиланням, якщо воно правильне, позначити користувача як підтвердженого.
	RQ 001.2.2. Якщо посилання неправильне, то вивести користувачу повідомлення про помилку.

2. Авторизація	<p>RQ 002.1.</p> <p>Система надає форму для авторизації з такими полями: email і пароль.</p>
----------------	--

Продовження таблиці 1.2

	<p>RQ 002.1.1.</p> <p>Якщо email введено в неправильному форматі, то система повідомляє користувачу про це.</p>
	<p>RQ 002.1.2.</p> <p>Якщо введений email ще не підтверджений, то на нього надсилається повторний лист-підтвердження реєстрації.</p>
	<p>RQ 002.1.3.</p> <p>Якщо аутентифікація пройшла, то система перенаправляє користувача у власний кабінет.</p>
	<p>RQ 002.1.4.</p> <p>Якщо користувач ввів неправильний пароль, то користувачу повідомляється про це та збільшується на 1 кількість невдалих спроб</p>
	<p>RQ 002.1.4.1.</p> <p>Після 10 невдалих спроб користувачу видається повідомлення з кількістю можливих</p>

	спроб(всього спроб 20) та рекомендацію відновити пароль
--	---

Продовження таблиці 1.2

	<p>RQ 002.1.4.2.</p> <p>Після 20 неправильної спроби ввести пароль система блокує аккаунт користувача та надсилає на електронну пошту лист з посиланням для розблокування аккаунту.</p>
	<p>RQ 002.1.5.</p> <p>Якщо користувача з такою електронною поштою в системі не існує, то користувачу виводиться повідомлення про це</p>
3. Скидання пароля	<p>RQ 003.1.</p> <p>Система надає користувачу форму для відновлення паролю з полем email</p>
	<p>RQ 003.1.1.</p> <p>Система перевіряє чи введений email в форматі електронної пошти</p>
	<p>RQ 003.2.</p> <p>На вказаний email користувачу надсилається лист з посиланням для скинення паролю</p>
	<p>RQ 003.2.1.</p> <p>При переході за посиланням система надає форму для скинення паролю з такими полями: email, пароль та підтвердження паролю</p>
	<p>RQ 003.2.1.1.</p>

	Система перевіряє чи введений email введено у форматі електронної пошти
	RQ 003.2.1.2. Система перевіряє чи введений email відповідає тому, на який було згенеровано посилання.

Продовження таблиці 1.2

	RQ 003.2.1.2.1. Якщо так, то система скидає пароль
	RQ 003.2.1.2.2. Якщо ні, то система виводить повідомлення про помилку
	RQ 003.2.1.3. Система перевіряє чи введений пароль співпадає з підтвердженням паролем
	RQ 003.2.1.4. Система перевіряє чи введений пароль більше 8 символів.
4.Розблокування облікового запису	RQ 004.1. При переході за посиланням з листа(RQ 002.1.4.2) система перевіряє його
	RQ 004.1.1. Якщо посилання правильне, то система розблокує аккаунт
	RQ 004.1.2. Якщо посилання неправильне система

	виводить повідомлення про помилку
	RQ 004.2. Система розблокує аккаунт автоматично після доби після блокування

Продовження таблиці 1.2

5.1. Зміна пароля	RQ 005.1 Система надає форму для зміни паролю з такими полями: старий пароль, новий пароль і підтвердження нового паролю
	RQ 005.1.1 Система має перевіряти пароль і підтвердження паролю, щоб вони співпадали і надавати користувачу повідомлення якщо це не так
	RQ 005.1.2 Система перевіряє чи введений новий пароль більше 8 символів і надавати користувачу повідомлення, якщо це не так
	RQ 005.1.3. Система перевіряє чи введений новий пароль правильний і якщо так, то змінює на новий. Інакше виводить повідомлення, що пароль

	невірний.
5.2. Зміна імені / прізвища	RQ 005.2. Система надає форму для зміни імені та прізвища

Продовження таблиці 1.2

6.1 Додавання і Аналіз знімку	RQ006.1 Система надає форму для Додавання і Аналіз знімку
6.2. Аналіз знімку	RQ006.2 Система перетворює Знімок в Результат Аналізу знімку
6.4 Додавання аналізованого знімку до бази	RQ006.4 Система зберігає знімок в кабінеті користувача
6.5 Перегляд списку Результатів аналізу знімків	RQ006.5 Система показує Користувачу Список Результатів Аналізу знімків

Таблиця 1.3 – Функціональні вимоги для актора Менеджер

7.1 Додавання патології	RQ007.1 Система надає форму для Додавання патології
7.2 Видалення патології	RQ007.2 Система видаляє патологію з бази даних
7.3 Перегляд списку	RQ007.3

патологій	Система Показує список патологій
7.4 Пошук патологій	RQ007.4 Система Показує фільтр списку патологій
8.1 Перегляд списку зразків захворювань	RQ008.1 Система Показує список зразків захворювань

Продовження таблиці 1.3

8.2 Пошук захворювань	RQ008.2 Система Показує фільтр списку зразків захворювань
8.3 Видалення зразку захворювання	RQ008.3 Система видаляє зразок захворювання з бази даних
8.4 Додавання зразку Захворювання	RQ008.4 Система надає форму для Додавання зразку захворювання
8.5 Редагування зразку захворювання	RQ008.5 Система надає форму для Редагування зразку захворювання
8.6 Редагування зразку захворювання	RQ008.6 Система надає форму для зв'язку захворювання з патологіями
9.1 Перегляд списку ресурсів	RQ009.1 Система Показує список ресурсів

9.2 Пошук ресурсів	RQ009.2 Система Показує фільтр списку ресурсів
9.3 Редагування ресурсу	RQ009.3 Система надає форму для Редагування ресурсу
9.4 Додавання ресурсу	RQ009.4 Система надає форму для Додавання ресурсу
9.5 Видалення ресурсу	RQ009.5 Система видаляє ресурс з бази даних

Таблиця 1.4 – Функціональні вимоги для актора Адміністратора.

Варіант використання	Функціональна вимога
10.1 Перегляд списку користувачів	RQ 010.1 Система надає Адміністратору список користувачів з такими полями: ім'я, електронна пошта, чи підтверджена електронна пошта, чи заблокований аккаунт, ролі
	RQ 010.1.1 Система додає до списку користувачів фільтр Пошуку користувача
10.2. Додавання користувача	RQ 010.2. Система надає А форму для створення нового користувача з полем email
	RQ 010.2.1. Поле email перевіряється чи правильний формат ел.пошти
	RQ 010.2.2. Система надсилає на заданий email лист-

	підтвердження з посиланням зі скиненням пароллю.
	RQ 010.2.2.1. Після того, як користувач скине пароль його пошта буде автоматично підтверджена.
10.4 Редагування користувачів	RQ 010.4. Система надає можливість редагувати користувача: його ролі, телефон, підтверджувати пошту, розблоковувати

Продовження таблиці 1.4

10.4.1. Зміна інформації користувача	RQ 010.4.1 Система надає форму редагування користувача
10.4.2 Зміна ролі користувача	RQ 010.4.2. Система надає можливість змінювати ролі до яких належить користувач(права доступу)
10.4.3 Підтвердження ел.пошти користувача	RQ 010.4.3. Система надає можливість підтверджувати ел.пошту користувача, якщо та не підтверджена
10.5. Розблокування Користувача	RQ 010.5 Система надає можливість розблокувати користувача, якщо він заблокований

Змн.	Арк.	№ докум.	Підпис	Дат

10.6. Фільтрувати користувача за назвою, електронною поштою , ролями , підтверджено , заблоковано	RQ 010.6.1. Система надає можливість сортувати користувачів по іменах в обидва боки

Продовження таблиці 1.4

	RQ 010.6.2. Система надає можливість відфільтровувати користувачів по підтверджених ел.поштах
	RQ 010.6.3. Система надає можливість відфільтровувати заблокованих користувачів
	RQ 010.6.4. Система надає можливість відфільтровувати користувачів по ролях
	RQ 010.6.5. Система надає можливість відфільтровувати користувачів по ролях
11.1 Перегляд списку ролей	RQ 011.1. Система надає можливість перегляду списку ролей

11.2. Додавання ролі	RQ 011.2. Система надає можливість створити роль
11.3. Видалення ролі	RQ 011.3. Система надає можливість видалити роль, крім системних ролей user і admin і менеджер

Продовження таблиці 1.4

11.4 Зміна дозволів ролі	RQ 011.4. Система надає можливість змінити права доступу для ролей. Є такі права доступу: право на перегляд списку користувачів (і на створення), на редагування користувачів, на редагування ролей користувачів, на редагування ролей
	RQ 011.4.1. За замовчування роль admin має всі права доступу і це змінити не можна.
	RQ 011.4.2. За замовчуванням роль user не має прав доступу, але це можна змінити.

1.3.4 Розроблення нефункціональних вимог

- Система повинна забезпечувати безпеку та конфіденційність даних користувачів.

- Система повинна бути відмово стійкою.
- Система має використовувати такий фреймворк для MVC – ASP.NET Core.
- Система має використовувати БД – MSSQL LocalDB Express, та таку технологію для роботи з нею – EntityFramework core, CodeFirst.
- Розгорнути на Azure.
- Система повинна забезпечувати надійність виконання запитів користувачів.
- Система має надавати інтуїтивно-зрозумілий інтерфейс користувача.
- Система повинна не накладати обмежень на кількість одночасно працюючих з нею користувачів.

На основі технічного завдання та з міркувань ергономіки програмного забезпечення були розроблена наступна постановка задачі:

- система має виконувати функції реєстрації/автентифікації користувачів;
- система має виконувати функцію цифрування і аналізу знімка;
- система має виконувати функцію управління результатами аналізу;
- система має виконувати функцію відображення статистики;
- система має виконувати функцію управління користувачами;
- система має виконувати функцію управління ролями та їх функціями(обмеженнями, правами доступу);
- система має виконувати функцію пошуку інформації по мітці або користувачу.

1.4 Висновки по розділу

У цьому розділі було наведено:

- опис предметної області;

- аналіз предметної області;
- аналіз існуючих технічних рішень;
- діаграма варіантів використання;
- опис варіантів використання;
- визначені функціональні вимоги та побудована їх діаграма;
- визначені нефункціональні вимоги.

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

У даному додатку є такі бізнес-процеси.

- Аутентифікація(Рисунок 2.1).
- Реєстрація(Рисунок 2.2).
- Відновлення паролю.
- Розблокування.
- Робота з користувачами.
- Робота з ролями.
- Пошук інформації.
- Робота користувача з власними даними.

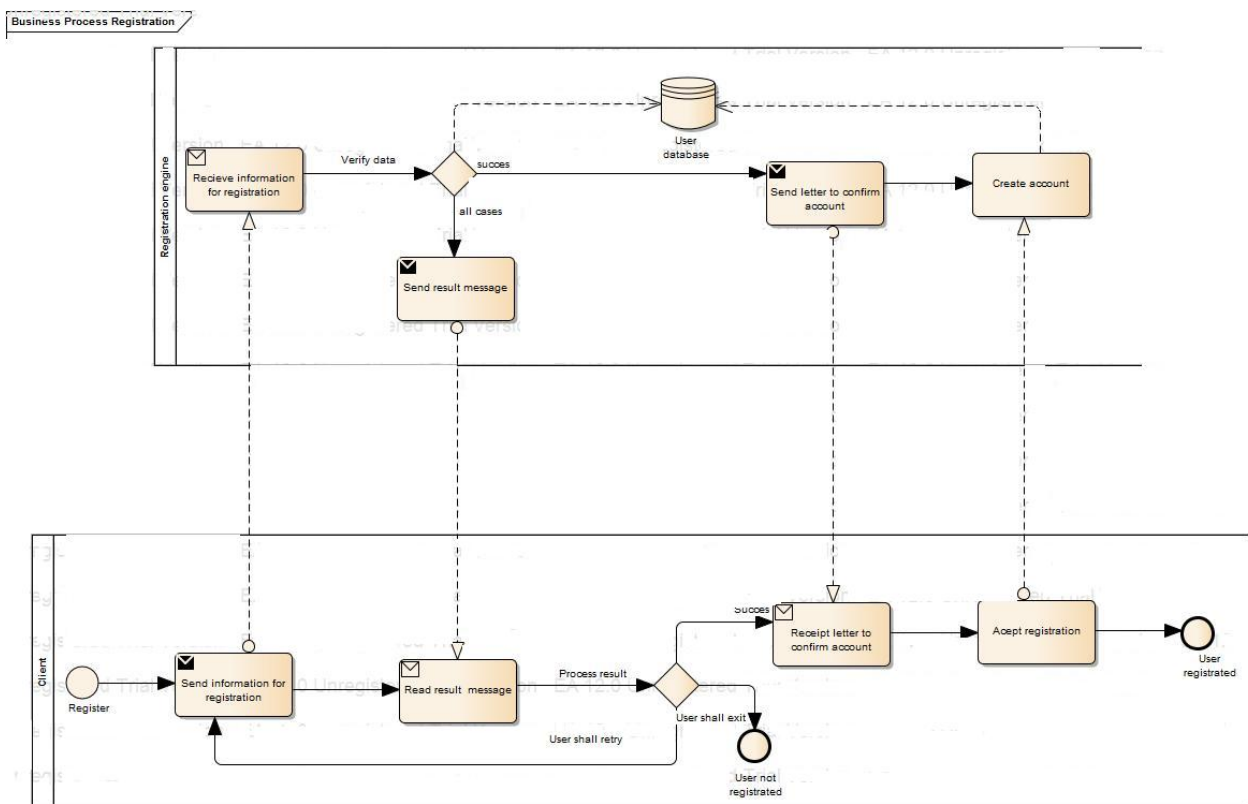


Рисунок 2.1 – Схема структурна бізнес-процесу реєстрації

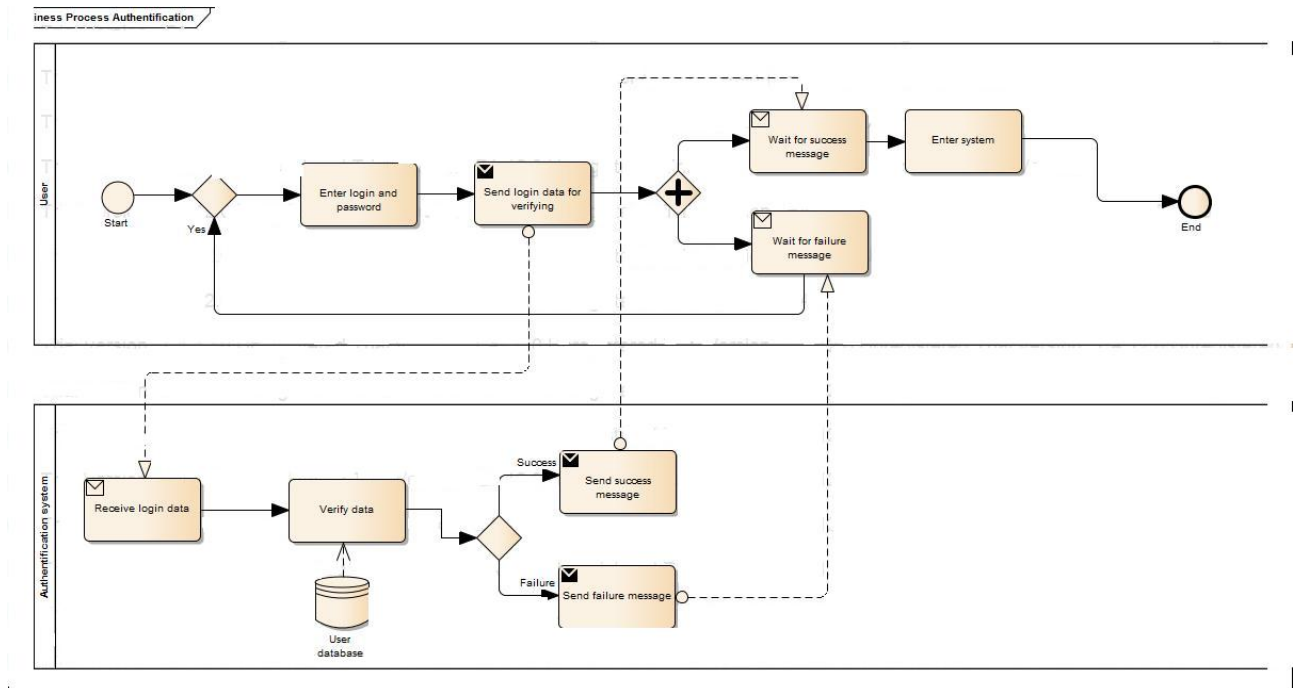


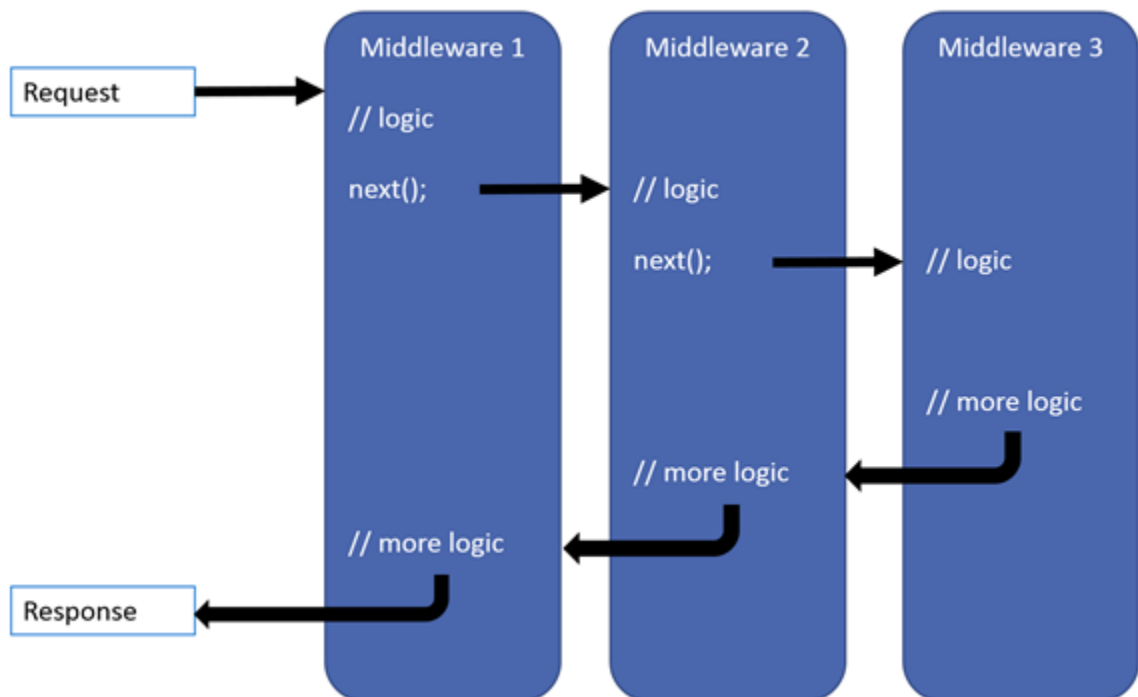
Рисунок 2.2 – Схема структурна бізнес-процесу аутентифікації

2.2 Архітектура програмного забезпечення

2.2.1 Платформа ASP.NET core

Платформа для створення Web-додатків від компанії Microsoft на мові C#. У цей час останньою версією цієї технології є ASP.NET Core. За допомогою цієї платформи можна створювати API, MVC, Empty та інші види проектів.

Конвеєр запитів ASP.NET Core складається з послідовності делегатів запиту, що викликаються один за іншим, як показано на Рисунку 2.3 (потік виконання позначений чорними стрілками).



Кожен з делегатів може виконувати операції до і після наступного делегата. Делегат також може прийняти рішення не передавати запит наступного делегату, що називається замиканням конвеєра запитів. Замикання часто є кращим, тому що дозволяє уникнути непотрібної роботи. Наприклад, компонент проміжного шару для статичних файлів може повернути запит статичного файлу і виконати замикання, щоб обійти решту конвеєра. Делегати обробки винятків повинні викликатися на початку конвеєра, щоб вони могли перехоплювати виключення, що виникають в більш пізніх стадіях.

2.2.2 Структура MVC

Структура архітектури MVC розділяє додаток на три основні групи компонентів: моделі, види і контролери. Це дозволяє реалізувати принципи поділу завдань. Згідно цій структурі запити користувачів направляються в контролер, який відповідає за роботу з моделлю для виконання дій користувачів і (або) отримання результатів

запитів. Контролер вибирає уявлення для відображення користувачеві з усіма необхідними даними моделі.

На рисунку 2.4 показані три основних компоненти і існуючі між ними зв'язки.

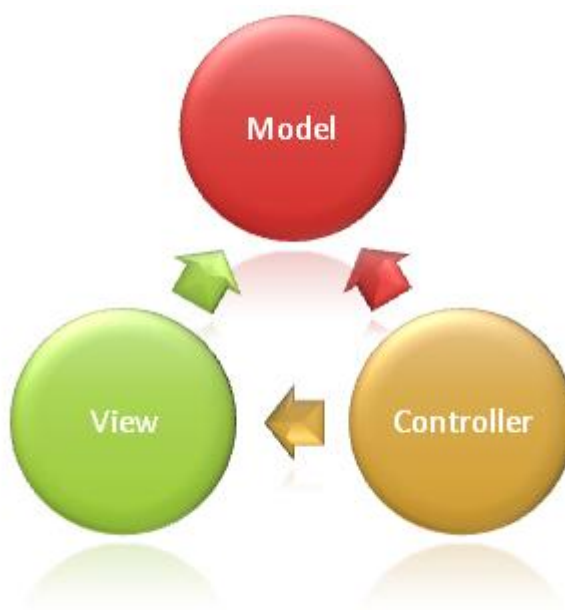


Рисунок 2.4 – MVC

Такий розподіл обов'язків дозволяє масштабувати додаток з точки зору складності, так як простіше писати код, виконувати налагодження та тестування компонента (моделі, уявлення або контролера), що має одне завдання (і наступного принципу єдиної обов'язки). Набагато важче оновлювати, тестувати і налагоджувати код, залежно якого знаходяться в двох або трьох цих областях. Наприклад, логіка призначеного для користувача інтерфейсу, як правило, піддається змінам частіше, ніж бізнес-логіка. Якщо код уявлення і бізнес-логіка об'єднані в один об'єкт, що містить бізнес-логіку, об'єкт необхідно змінювати при кожному оновленні призначеного для користувача інтерфейсу. Це часто призводить до виникнення помилок і необхідності повторно тестувати бізнес-логіку після кожного незначного зміни призначеного для користувача інтерфейсу.

Model – моделі. Використовуються для відображення моделі предметної області в структурах даних. Частіше за все через моделі за допомогою EntityFramework core здійснюється робота з БД. EntityFramework core є можливість підходу CodeFirst, тобто спочатку пишеться код (структуру даних), а потім код автоматично конвертується у таблиці і зв'язки у базі даних. При такому підході, для зміни структури БД використовуються міграції, про які буде нижче.

View – представлення, HTML сторінки, тобто те, що буде відображатися у нас на сторінках. У ASP.NET core використовується звучний двигун для представлень – Razor. За допомогою нього можна робити вкраплення C# коду, що є дуже зручним. Вкраплення починаються з @ та мають бути у фігурних дужках {}. Детальніше про представлення буде далі.

Controller – контролери.

Центральною ланкою в архітектурі ASP.NET Core MVC є контролер. При отриманні запиту система маршрутизації обирає для обробки запиту потрібний контролер і передає йому дані запиту. Контролер обробляє ці дані і посилає назад результат обробки.

У контролерах знаходиться логіка застосунку.

2.2.3 Схема БД та опис таблиць

Схема БД представлена на в листі 3, додатку В.

У Таблиці 2.1 наведено опис таблиць. Умовні позначення:

- а) [Index] – індекс таблиці;
- б) [PK] – первинний ключ;
- В) [FK] – зовнішній ключ.

Таблиця 2.1 – Опис таблиць

AspNetUsers	
Id [PK]	Ідентифікатор
AccessFailedCount	Кількість невдалих спроб ввести пароль
Email	Електронна пошта
EmailConfirmed	Чи підтверджена електронна пошта
LockoutEnabled	Чи дозволено блокування користувача
LockoutEnd	Кінець блокування
NormalizedEmail [Index]	Електронна пошта у верхньому регістрі
NormalizedUserName [Index]	UserName у верхньому регістрі
PasswordHash	Хеш паролю
PhoneNumber	Номер телефону
PhoneNumberConfirmed	Чи підтверджений номер телефону
SecurityStamp	Сіль
TwoFactorEnabled	Чи підключена двухфакторна аутентифікація
UserName	UserName
Name	Ім'я
Surname	Прізвище
RegistrationDate	Дата реєстрації
LastActivityDate	Дата останньої активності
AspNetRoles	
Id[PK]	Ідентифікатор
NormalizedName[Index]	Ім'я в верхньому регістрі
Name	Ім'я
ConcurrencyStamp	Узгодження зв'язку
AspNetRoleClaims	

Id[PK]	Ідентифікатор
--------	---------------

Продовження таблиці 2.1

ClaimType	Тип поля
ClaimValue	Значення поля
RoleId	Ідентифікатор ролі
AspNetUserLogins	
LoginProvider [FK]	Логін провайдера
ProviderKey[FK]	Ключ провайдера
ProviderDisplayName	Найменування провайдера
UserId	Ідентифікатор користувача
AspNetUserRoles	
UserId [PK]	Ідентифікатор користувача
RoleId [PK]	Ідентифікатор ролі
AspNetUserTokens	
UserId [PK]	Ідентифікатор користувача
LoginProvider [PK]	Ідентифікатор провайдера
Name [PK]	Ім'я токєну
Value	Значення токєну
EKGPatology	
Id [PK]	Ідентифікатор
Name	Найменування
Description	Опис
Start	Стартова позиція на ресурсі
[End]	Кінцева позиція на ресурсі
ResourceId	Ідентифікатор ресурсу
PatologyId	Ідентифікатор патології
EKGSampleId	Ідентифікатор зразка захворювання
EKGSample	

Продовження таблиці 2.1

Id [PK]	Ідентифікатор
Name	Найменування
SourceId	Ідентифікатор ісходного аналізу малюнка
IntervalsID	Ідентифікатор результату аналізу малюнка
Dm	Розмірність результату
SourceDm	Розмірність ісходного аналізу малюнка
ResourceId	Ідентифікатор ресурсу
AnalysisPosition	позиція
Patology	
Id [PK]	Ідентифікатор
Name	Найменування
Description	Опис
Resource	
Id [PK]	Ідентифікатор
Name	Найменування
Path	Шлях до файлу
Snapshot	
Id [PK]	Ідентифікатор
SourceId	Ідентифікатор ісходного аналізу малюнка
IntervalsID	Ідентифікатор результату аналізу малюнка
ResourceId	Ідентифікатор ресурсу
UserId	Ідентифікатор користувача
CreatedOn	Дата створення
SampleId	Ідентифікатор зразка захворювання
Pos	Позиція в результаті аналізу малюнка
SourceDm	Позиція в ісходному аналізу малюнка

Sources

Продовження таблиці 2.1

Id [PK]	Ідентифікатор
Letters	Строка

У Таблиці 2.2 наведено опис класів. Схему структурну класів можна подивитися у листі 2, додатку В.

Таблиця 2.2 – Опис класів

AccountController	
ISender _emailSender;	Сервіс для надсилання повідомлень
UserManager<ApplicationUser> _userManager	Сервіс для управління користувачами
ILogger _logger	Сервіс для логування
SignInManager<ApplicationUser> _signInManager	Сервіс для аутентифікації користувача
ApplicationDbContext db	Контекст БД
AccountController(UserManager<ApplicationUser> userManager, SignInManager<ApplicationUser> signInManager, ISender emailSender, ILoggerFactory loggerFactory, ApplicationDbContext context, IGetUserInfo)	Конструктор

Продовження таблиці 2.2

<pre>public async Task<IActionResult> Login(LoginViewModel model, string returnUrl = null)</pre>	<p>POST-запит аутентифікації</p> <p>Приймає: LoginViewModel model, string returnUrl = null</p> <p>Повертає: якщо все правильно, то перенаправляю в кабінет користувача, інакше повертає форму Login з помилками</p>
<pre>public async Task<string> GetConfirmationCode(string userId)</pre>	<p>Приймає: string userId</p> <p>Повертає: код підтвердження</p>
<pre>public IActionResult Register(string returnUrl = null)</pre>	<p>GET-запит на форму реєстрації</p> <p>Приймає: string returnUrl</p> <p>Повертає: форму реєстрації</p>

Продовження таблиці 2.2

public async Task<IActionResult> Register(RegisterViewModel model, bool isAdminCreate, string returnUrl = null)	POST-запит реєстрації Приймає: RegisterViewModel model, bool isAdminCreate, string returnUrl = null Повертає: форму ConfirmEmail якщо все правильно
public async Task<IActionResult> LogOff()	POST-запит «вийти з аккаунту» Повертає: форму Login
public async Task<ActionResult> ConfirmUnlock(string UserId, string Code)	GET-запит Приймає: string UserId, string Code Повертає: Форму Login
public async Task<IActionResult> ConfirmEmail(string userId, string codeConfirm, string codeReset)	GET-запит Приймає: string userId – ідентифікатор користувача, string codeConfirm – код підтвердження, string codeReset – код скинення паролю Повертає: форму скинення паролю

Продовження таблиці 2.2

public IActionResult ForgotPassword()	GET-запит Повертає: форму ForgotPassword
public async Task<IActionResult> ForgotPassword(ForgotPasswordViewModel model)	POST-запит Приймає: ForgotPasswordViewModel model Повертає: якщо все правильно повідомлення, щоб користувач перевірів свою пошту
public IActionResult ResetPassword(string code = null)	GET-запит Приймає: string code Повертає: форму ResetPassword
public async Task<IActionResult> ResetPassword(ResetPasswordViewModel model)	POST-запит Приймає: ResetPasswordViewModel model Повертає: форму ResetPasswordConfirmation якщо все пройшло успішно, інакше форму ResetPassword

Продовження таблиці 2.2

private void AddErrors(IdentityResult result)	Додає помилки до стану моделі, що виникли в результаті виконання операцій з користувачами Приймає: IdentityResult result – результат роботи з UserManager
private Task<ApplicationUser> GetCurrentUserAsync()	Повертає: користувача цієї сесії
private IActionResult RedirectToLocal(string returnUrl)	Приймає: returnUrl – посилання для перенаправлення Повертає: перенаправляє на локальну сторінку зазначені у вхідному параметрі
RolesController	
RoleManager<IdentityRole> _roleManager	Сервіс для роботи з ролями
IFunctionChecker _checker;	Сервіс для перевірки приналежності до функції
RolesController(RoleManager<IdentityRole> roleManager, IFunctionChecker checker)	Конструктор
public IActionResult Index()	Метод, що повертає

форму з ролями

Продовження таблиці 2.2

public async Task<ActionResult> Create(string role)	Метод створення ролі
public async Task<ActionResult> Delete(string name)	Метод видалення ролі
public async Task<ActionResult> Save(string role, string claim, bool wasChecked, string url)	Метод збереження налаштувань функцій
UsersController	
RoleManager<IdentityRole> _roleManager;	Сервіс для роботи з ролями
UserManager<ApplicationUser> _userManager;	Сервіс для роботи з користувачами
IGetUserInfo _userInfo;	Сервіс для отримання username та генерації BleMasterKey
ApplicationDbContext db;	Контекст БД
public UsersController(RoleManager<IdentityRole> roleManager, userManager<ApplicationUser> userManager, ApplicationDbContext _db, IGetUserInfo userInfo)	Конструктор
public async Task<ActionResult> Index(int? page, string seq, string role, bool direction, string locked, string confirmed,	Виведення списку користувачів, приймає номер сторінки, вираз для пошуку та фільтри

string name,

Продовження таблиці 2.2

private async Task<List<UserForAdmin>> Search(string seq)	Пошук користувача, приймає вираз для пошуку
private List<UserForAdmin> FilterRoles(List<UserForAdmin> list, string role, bool direction)	Фільтрація користувачів по ролях
public async Task<IActionResult> Unlock(string id)	Розблокування обраного користувача, приймає ідентифікатор користувача
public async Task<IActionResult> ConfirmEmail(string id)	Підтвердження електронної пошти користувача адміном, приймає ідентифікатор користувача
public async Task<IActionResult> Edit(string id)	Повертає форму редагування користувача, приймає ідентифікатор користувача
public IActionResult Create()	Повертає форму створення користувача
private string RandomGenerator(int length)	Генерування рандомного пароллю(необхідно при створенні користувача з адмінки)

Продовження таблиці 2.2

public async Task<ActionResult> Edit(string PhoneNumber, string id, Dictionary<string,bool> dict)	POST-запит Редагування користувача Приймає: телефонний номер, ідентифікатор, пари ключів по ролях до яких користувач
public async Task<ActionResult> Delete(string id)	Видалення користувача

2.2.4 Migrations

Як зазначалося вище у ASP.NET MVC використовується EntityFramework & CodeFirst. Отож коли необхідно змінити щось у структурі БД, наприклад, додати стовпчик у таблицю достатньо просто прописати його у відповідній моделі. Потім у консолі диспетчера пакетів прописати(Рисунок 2.5):

Add-Migration Name

А потім Update-Database

Також це можна зробити через cmd та команди dotnet

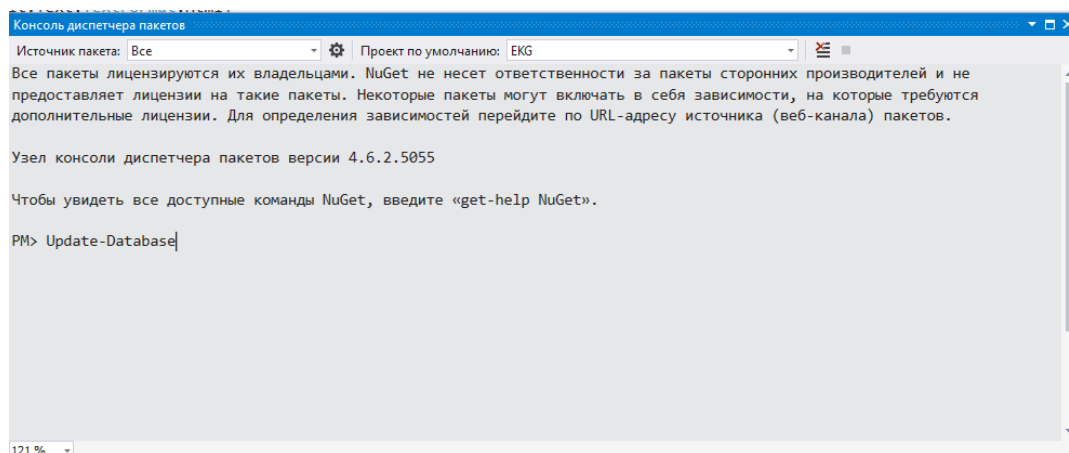


Рисунок 2.5 – Оновлювання бази даних

При першому використанні проекту або коли з файлом БД щось стається необхідно виконати команду Update-Database.

Під час виконання цієї команди спочатку створюються стандартні таблиці(міграція Initial_Create), а потім виконуються всі міграції, що ми створили раніше. Таким чином це дуже зручно для застосування у проектах над якими працює команда.

2.2.3 Models

Анотації для моделей

```
[EmailAddress(ErrorMessageResourceType = typeof(Resources.Resource),
    ErrorMessageResourceName = "email_format")]
```

Підключає валідацію для формату email

ErrorMessageResourceType – задає тип ресурсу

ErrorMessageResourceName – задає ключ для ресурсу

```
[Required(ErrorMessageResourceType = typeof(Resources.Resource),
    ErrorMessageResourceName = "email_req")]
```

Елемент обов'язків для введення

```
[Display(Name = "email", ResourceType =
    typeof(Resources.Resource))]
```

Задає ім'я відображення

```
DataType(DataType.Password)]
```

Задає тип пароллю

```
StringLength(100, ErrorMessageResourceType =
    typeof(Resources.Resource),
    ErrorMessageResourceName = "pas_length",
    MinimumLength = 8)]
```

Задає необхідну для введення довжину поля

```
[Compare("Password", ErrorMessageResourceType
typeof(Resources.Resource),
ErrorMessageResourceName = "error_compare_pas")]
```

2.3 Аналіз безпеки даних

При розробці сервісу безпеки було використано бібліотеку Asp.NET Core 2.0 WebApi JWT

JSON Web Token (JWT) - це відкритий стандарт (RFC 7519), який визначає компактний та автономний спосіб безпечного передавання інформації між сторонами як об'єкт JSON. Цю інформацію можна перевірити та довірити, оскільки вона підписана цифровим способом. JWT можна підписати, використовуючи таємницю (з алгоритмом HMAC) або пару публічних / приватних ключів, використовуючи RSA або ECDSA .

Хоча JWT можна зашифрувати, щоб забезпечити таємницю між сторонами, ми зосередимось на підписаних токенах. Токени, що підписані, можуть перевіряти цілісність претензій, що містяться в ньому, тоді як зашифровані токени приховують ці претензії від інших сторін. Коли токени підписуються з використанням пар із загальнодоступними / приватними ключовими словами, підпис також засвідчує, що тільки сторона, яка має приватний ключ, є тими, хто її підписав.

Ось кілька сценаріїв, де корисні JSON Web Tokens:

Авторизація : це найпоширеніший сценарій використання JWT. Після входу в систему кожен наступний запит включає JWT, що дозволяє користувачеві отримувати доступ до маршрутів, служб та ресурсів, дозволених за допомогою цього токена. Система єдиного входу - це функція, яка сьогодні широко використовує JWT, через її невеликі накладні витрати та можливість легко користуватися різними доменами.

Обмін інформацією: JSON Web Tokens - хороший спосіб надійно передавати інформацію між сторонами. Через те, що JWT можна

підписати, наприклад, використовуючи пари публічних / приватних ключів, ви можете бути впевнені, що відправники - це ті, хто вони кажуть. Окрім того, оскільки підрахунок розраховується за допомогою заголовка та корисної інформації, ви також можете перевірити, чи вміст не було змінено.

Як працюють JSON Web Tokens?

При аутентифікації, коли користувач успішно входить до системи, використовуючи свої облікові дані, буде повернуто веб-токен JSON. Оскільки маркери є верифікаторами, велику увагу слід приділяти запобіганню проблемам безпеки. Загалом, ви не повинні зберігати токени довше, ніж потрібно.

Кожного разу, коли користувач хоче отримати доступ до захищеного маршруту або ресурсу, агент користувача повинен надіслати JWT, як правило, у заголовку авторизації, використовуючи схему носіїв. Зміст заголовка повинен виглядати наступним чином:

Authorization: Bearer <token>

Це може бути, у певних випадках, механізмом авторизації без паспорта. Захищені маршрути сервера перевірятимуть наявність дійсного JWT в `Authorization` заголовку, і, якщо він присутній, користувачу буде надано доступ до захищених ресурсів. Якщо JWT містить необхідні дані, потреба в запиті до бази даних для певних операцій може бути зменшена, хоча це може бути не завжди можливим.

Якщо токен надсилається у `Authorization` заголовку, спільне використання ресурсів спільного користування (CORS) не буде проблемою, оскільки вона не використовує файли cookie.

Рисунок 2.6 показує, як отримано та використовується JWT для доступу до API або ресурсів:

					КПІ-ІП-34111.045490-02-81	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дат		

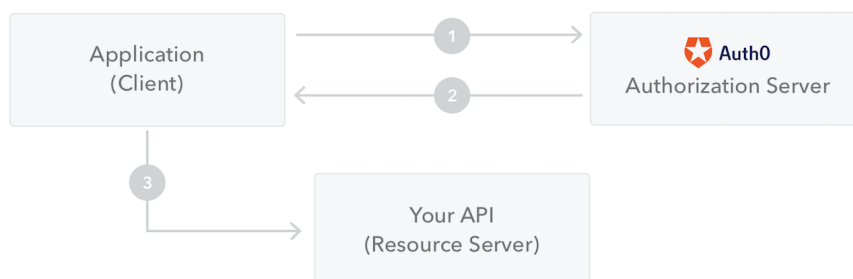


Рисунок 2.6 – JWT для доступу до API або ресурсів

Заявка або клієнт запитує авторизацію серверу авторизації. Це виконується за допомогою одного з різних авторизаційних потоків. Наприклад, типова веб-програма, сумісна з OpenID Connect, пройде через `/oauth/authorize` кінцеву точку за допомогою потоку коду авторизації.

Коли авторизація надається, сервер авторизації повертає маркер доступу до програми.

Програма використовує маркер доступу для доступу до захищеного ресурсу (наприклад, API).

Висновки по розділу

У даному розділі було проаналізовано, створено та наведено:

- архітектура ПЗ;
- діаграму класів та їх опис;
- схему БД та опис таблиць та їх атрибутів;
- опис використаних підходів до написання програмного коду з

прикладом коду.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Перелік юніт-тестів для основного функціоналу

У комп'ютерному програмуванні юніт-тестування – це метод тестування ПЗ, за допомогою якого тестуються окремі одиниці вихідного коду, набори одного або кількох програмних модулів разом із відповідними контрольними даними, процедурами користування та робочими процедурами для визначення їх придатності для використання [5].

При розробці даного ПЗ було вирішено покрити юніт-тестами частину функціоналу аналізу ЕКГ, що відповідає за математичні обрахунки. По-перше, це забезпечить надійність роботи при будь-яких змінах у коді – знайти помилку у новому або зміненому коді програми буде значно легше і швидше. По-друге, це може у якійсь мірі замінити або доповнити вже готову документацію, оскільки кожен юніт-тест має опис, що саме у ньому тестується. По-третє, якщо розробка ведеться через тестування (TDD-підхід), це означає, що тести відображають саме бізнес-суть і не є формальністю.

Перелік функціоналу, під який були написані юніт-тести:

- визначення лексемної букви у випадку, якщо графік ЕКГ виходить за грань картинки (A – зверху, Z – знизу);
- визначення шагу лексемної букви за амплітудою;
- обрахунок масиву колонок пікселів із знімку ЕКГ;
- обрахунок масиву рядків пікселів із знімку ЕКГ;
- обрахунок схожості двох пікселів із заданим відхиленням;
- знаходження найбільш темних та найбільш світлих пікселів у масиві;

- знаходження середнього індексу при декількох однакових пікселях;
- обрахунок лексемних букв за заданими індексами;
- обрахунок розміру однієї ЕКГ-клітинки у пікселях;
- блокування кнопки входу при невалідних даних для входу;
- згортання/розгортання результатів даних ЕКГ в залежності від їх наявності;
- перевірка локалізації, поточної мови;
- згортання результатів даних ЕКГ у випадку помилки.

Усього за період розробки було написано близько п'яти дюжин юніт-тестів, запуск яких в сумі займає в середньому 10 секунд. Графічний інтерфейс, у якому тести запускалися, показаний на рисунку 3.1.

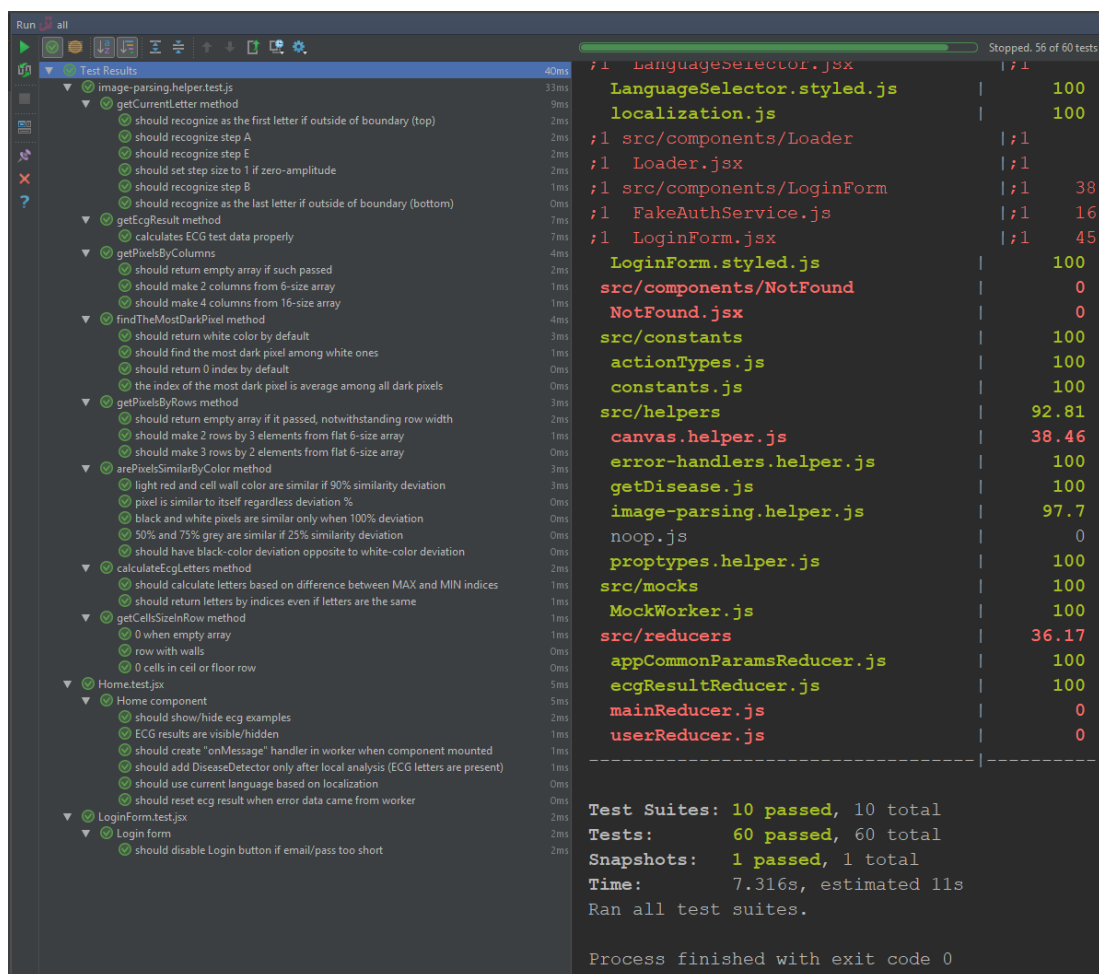


Рисунок 3.1 – Графічний інтерфейс для запуску юніт-тестів

КПІ-ІП-34111.045490-02-81

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Програмне забезпечення може бути розгорнуте на будь-якому типі персональних комп'ютерів: десктоп, ноутбук, планшет, смартфон.

4.1.1 Мінімальна конфігурація технічних засобів

Тип процесору: Intel Core i3+.

Об'єм ОЗП: 1Гб.

Об'єм фізичної пам'яті: 64Гб.

4.1.2 Вимоги до операційних систем

Не принципово, всі обчислювальні операції виконуються у браузері (кросс-платформенно) та на серверній частині. На пристрій користувача надходять лише результативні дані.

4.1.3 Вимоги до інформаційної та програмної сумісності

Для того, щоб зайти на сайт необхідно мати браузер на пристрої, найкраще Chrome, Opera, Firefox. Сайт також підтримує IE 11 та Safari.

4.1.4 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

4.1.5 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

4.1.6 Спеціальні вимоги

Спеціальні вимоги не висуваються.

4.1.7 Публікація програмного забезпечення

Для публікації необхідно у контекстному меню рішення проекту знайти «Опубликовать/Publish». Після цього відкриється форма як на Рисунку 4.1

Рисунок 4.1 – Перша форма публікації

Якщо додаток публікується на Azure, то створюєте веб-додаток на azurewebsites, вказуєте строку підключення до БД(частіше за все там же на Azure). Після цього можна там завантажити профіль публікації і імпортувати його на формі вище.

Після того, як ви вибрали профіль відкривається друга форма(Рисунок 4.2). На цій формі вказується сервер, ім'я сайту, ім'я користувача, пароль та цільовий url.

Рисунок 4.2 – Друга форма публікації

Далі відкривається третя форма публікації «Параметри»(Рисунок 4.3).

На цій сторінці конфігуруються рядки підключення до БД, підхід до міграцій, політика поведінки з файлами на сервері.

Після цього можна проглянути зміни, які будуть публікуватися на сервері.

Після цього, якщо все було виконано правильно, додаток опублікується.

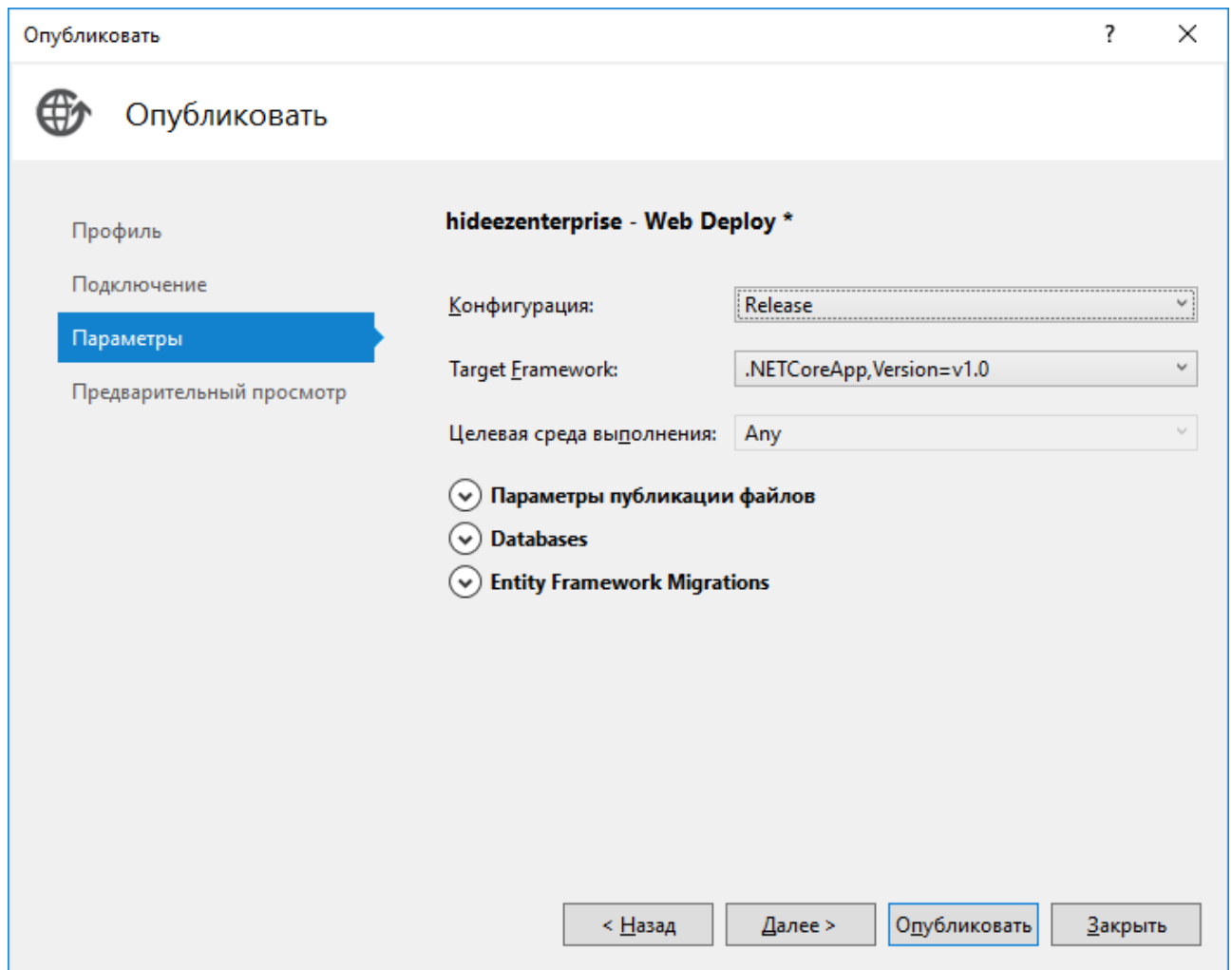


Рисунок 4.3 –Третья форма публікації

4.2 Работа з програмним забезпеченням

Керівництво користувача наведено в документі «Керівництво користувача».

ВИСНОВКИ

Було розроблено Web-сервіс для прискорення процесу аналізу ЕКГ.
Усі функціональні вимоги було виконано і протестовано.

Було вивчено, засвоєно та використано:

- принцип обробки запиту та маршрутизації;
- принцип роботи MVC;
- принцип роботи CodeFirst, EntityFramework та міграцій;
- принципи побудови адаптивного дизайну;
- надіслання Emails;
- створення мультимовного сайту, робота з ресурсами,

локалізація моделей та повідомлень валідації;

- використання Razor двигуна;
- принципи TDD, розробки через тестування;
- принципи публікації веб-застосунку до хмарного сховища;
- розробка технічної документації за ДСТУ;
- описано модель комплексу програм для розбору ЕКГ та

проаналізовано різні лексемні моделі;

- наведені міри оцінки для визначення якості прогнозу на часових рядах при застосуванні лінгвістичного моделювання;
- розглянути можливості подання часових рядів з фрактальними властивостями у вигляді лінгвістичних моделей для подальшого аналізу модельованого динамічного процесу.

Планується застосування запропонованого підходу для аналізу та прогнозування часових рядів різної природи.

РОЗРОБКА ТЕХНІЧНОЇ ДОКУМЕНТАЦІЇ ЗА ДСТУ.ПЕРЕЛІК ПОСИЛАНЬ

- 1) Односторінкові веб-застосунки [Електронний ресурс] - Режим доступу до ресурсу: <https://msdn.microsoft.com/en-us/magazine/dn463786.aspx>
- 2) Онлайн уроки по ASP.NET MVC [Електронний ресурс] – Режим доступу до ресурсу: <http://metanit.com/sharp/mvc.php>
- 3) Статті по ASP.NET MVC [Електронний ресурс] – Режим доступу до ресурсу: <https://habrahabr.ru/post/175999/>
- 4) Конвертація типів C# у типи MSSQL [Електронний ресурс] – Режим доступу до ресурсу: <http://stackoverflow.com/questions/425389/c-sharp-equivalent-of-sql-server-datatypes>
- 5) Робота з EntityFramework [Електронний ресурс] – Режим доступу до ресурсу: <https://msdn.microsoft.com/ru-RU/data/jj591621.aspx>
- 6) Хешування паролів [Електронний ресурс] – Режим доступу до ресурсу: <http://dotnet.today/ru/aspnet5-vnext/security/data-protection/consumer-apis/password-hashing.html>
- 7) Публікація на azure+git [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/azure/app-service-web/>
- 8) Перевірка достовірності на клієнті [Електронний ресурс] – Режим доступу до ресурсу: https://professorweb.ru/my/ASP_NET/mvc/level7/7_6.php
- 9) Керівництво по конфігурації [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/configuration>
- 10) Робота з cookie-файлами [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/aspnet/core/security/authentication/cookie>

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ Павлов О.А.

“ ” _____ 2019 р.

Комплекс програм для віддаленого доступу до бази знань патернів
аномалій роботи серця людини

Технічне завдання

КПІ-ІІ-34111.045490-03-91

“ПОГОДЖЕНО”

Керівник проекту:

Баклан І.В.

Нормоконтроль:

Ліщук К.І.

Виконавець:

Сорочинський Р. В.

Київ – 2019 року

ЗМІСТ

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ	3
2 ПІДСТАВА ДЛЯ РОЗРОБКИ.....	4
3 ЦІЛІ РОЗРОБКИ.....	5
4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	8
6 СТАДІЇ І ЕТАПИ РОЗРОБКИ	9
7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ	10

					КПІ-ІІІ-34111.045490-03-91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: «Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини»

Галузь застосування:

Наведене технічне завдання поширюється на розробку застосування для аналізу фотографії знімка електрокардіограми серця користувача аналізу фотографії знімка електрокардіограми серця користувача на базі збережених патернів, котре використовується для ранньої діагностики роботи серця.

					КПІ-ІП-з4111.045490-03-91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки “Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини”, є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» (НТУУ «КПІ ім. Сікорського»).

					КТП-ІП-34111.045490-03-91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ЦІЛІ РОЗРОБКИ

- Дослідити існуючі методи аналізу діагностичних ознак ЕКГ.
- Розробити алгоритм аналізу фотографії знімка електрокардіограми серця.
- Розробити програмний модуль.
- Протестувати розроблений програмний модуль на модельних сигналах.
- Оформити рекомендації щодо супроводу програмного продукту.

Ціль розробки це покращення швидкості діагностики роботи серця.

Метою створення розробки є засоби для зменшення витраченого часу для ранньої діагностики роботи серця.

Призначення розробки:

- аналіз знімку електрокардіограми;
- додавання до бази знань патернів;
- редагування бази знань патернів;
- редагування списку патологій;
- відображення результату аналізу.

					КПІ-ІП-з4111.045490-03-91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

Для користувача:

аналіз фотографії знімка електрокардіограми серця;

показ результату аналізу;

Для менеджера бази патернів

Додавання до бази знань патернів;

редагування бази знань патернів

Система повинна забезпечувати безпеку та конфіденційність даних користувачів.

Система повинна бути відмово стійкою.

Система має використовувати такий фреймворк для MVC – ASP.NET Core.

Система має використовувати БД – MSSQL LocalDB Exspress, та таку технологію для роботи з нею – EntityFramework core, CodeFirst.

Система повинна забезпечувати надійність виконання запитів користувачів.

Система має надавати інтуїтивно-зрозумілий інтерфейс користувача.

Система повинна не накладати обмежень на кількість одночасно працюючих з нею користувачів.

4.1.2 Вимоги до організації вхідних даних

4.1.3 Розробку виконати на платформі ASP.NET Core та за допомогою бібліотеки React

4.2 Вимоги до надійності

					КТП-ІП-з4111.045490-03-91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2.1 Передбачити контроль введення інформації.

4.2.2 Передбачити захист від некоректних дій користувача.

4.3 Умови експлуатації

4.3.1 Умови експлуатації

Не висуваються.

4.3.2 Обслуговуючий персонал не використовується.

4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Програмне забезпечення повинно функціонувати на IBM-сумісних персональних комп'ютерах.

4.4.2 Мінімальна конфігурація технічних засобів

4.4.2.1 Тип процесору: Intel Core i3+;

4.4.2.2 Об'єм ОЗП: 1 ГБ;

4.4.2.3 Об'єм фізичної пам'яті: 64Гб ;

Вимоги до інформаційної та програмної сумісності

Не висуваються.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

4.8 Спеціальні вимоги

Згенерувати установочну версію програмного забезпечення.

					КПІ-ІП-з4111.045490-03-91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Попередній склад програмної документації.

- Супроводжувальна документація
- Пояснювальна записка.
- Керівництво користувача.
- Довідникова документація
- Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.
- Графічна документація.
- Схема структурна послідовності.
- Схема структурна варіантів використання.
- Схема бази даних.
- Схема структурна станів інтерфейсу.
- Креслення вигляду екранних форм.

					КТП-ІП-34111.045490-03-91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення літератури за тематикою проекту		
2.	Розробка технічного завдання		Технічне завдання
3.	Аналіз вимог та уточнення специфікацій		Специфікації програмного забезпечення
4.	Проектування структури програмного забезпечення, проектування компонентів		Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму ...)
5.	Програмна реалізація програмного забезпечення		Тексти програмного забезпечення
6.	Тестування програмного забезпечення		Тести, результати тестування
7.	Розробка матеріалів текстової частини проекту		Пояснювальна записка.
8.	Розробка матеріалів графічної частини проекту		Графічний матеріал проекту
9.	Оформлення технічної документації проекту		Технічна документація

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ**7.1 Види випробувань**

Тестування розробленого програмного продукту виконується відповідно до “Аналіз якості та тестування програмного забезпечення”.

					КТП-ІП-34111.045490-03-91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ Павлов О.А.

“ ____ ” _____ 2019 р.

**Комплекс програм для віддаленого доступу до бази знань патернів
аномалій роботи серця людини**

Керівництво користувача

КПІ-ІП-34111.045490-05-34

“ПОГОДЖЕНО”

Керівник проекту:

_____ Баклан І.В.

Нормоконтроль:

_____ Лішук К.І

Виконавець:

_____ Сорочинський Р.В.

Київ – 2019 року

ЗМІСТ

1 ЗАГАЛЬНІ ВІДОМОСТІ.....	3
2 ПІДГОТОВКА ДО РОБОТИ	4
3 ПОЧАТОК РОБОТИ.....	5

					КПІ-ІП-34111.045490-05-34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНІ ВІДОМОСТІ

У даній роботі була розглянута задача збору і аналізу даних ЕКГ. Ця задача вирішує проблему оперативності такого аналізу і значно збільшує його швидкість. До того ж, вона може бути корисна в регіонах з відсутністю або малою кількістю лікарів, тому що кожен бажаючий зможе перевірити показання ЕКГ свого серця зі смартфона сам.

Система надає користувачу різний інтерфейс в залежності від його ролі, після чого він може або завантажити знімок ЕКГ на аналіз, або завантажити патерн ЕКГ як патологію, або переглянути список інших користувачів із можливістю редагування їхніх ролей.

					КПІ-ІП-34111.045490-05-34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПІДГОТОВКА ДО РОБОТИ

2.1 Системні вимоги для коректної роботи

Веб-застосунок розроблено для запуску на нових браузерях. Для функціонування застосунку потрібно мати мобільний телефон, планшет чи ПК. Оскільки даний ПП був розроблений із використанням підходу MobileFirst, тут і надалі ілюстрації будуть наведені саме під мобільні розширення екрану.

Системні вимоги:

- мінімальне розширення екрану 480×800;
- об'єм ПЗП – 30 МВ;
- об'єм ОЗП – 128 МВ.

2.2 Запуск веб-застосунку

На пристрої необхідно мати встановлений браузер найновішої версії: Google Chrome, Mozilla Firefox, Opera, Internet Explorer 11, Edge або Safari. Щоб запустити застосунок, достатньо перейти за посиланням <https://goodwin64.github.io/din-don-heart>, застосунок буде запущено автоматично.

2.3 Перевірка коректної роботи

Після першого запуску застосунку користувач має побачити вікно авторизації – поля для входу із вводом електронної скриньки та паролю. У такому випадку застосунок було запущено коректно.

					КПІ-ІП-34111.045490-05-34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ПОЧАТОК РОБОТИ

Для того, щоб перейти на головну сторінку, потрібно авторизуватися на сайті – ввести ел. скриньку та пароль.

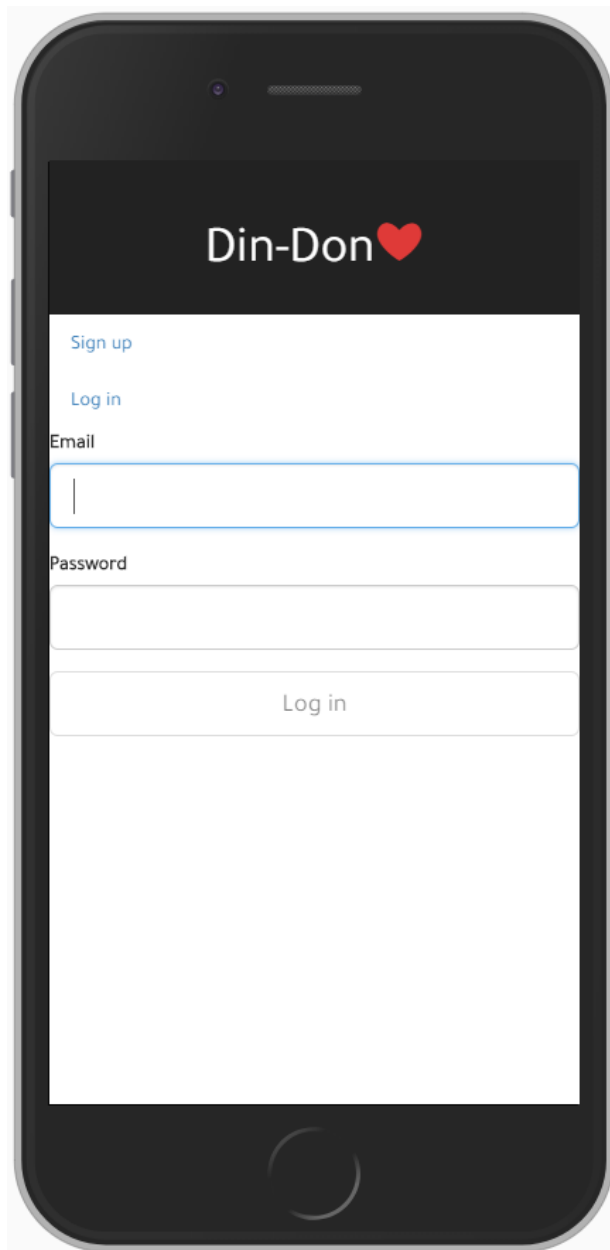


Рисунок 3.1 – Сторінка входу

Далі, якщо роль користувача - «Користувач» (потенційно хвора людина, котра хоче перевірити свою кардіограму), його перенаправлено на головну сторінку. Серед доступних функцій – «вийти», «змінити мову», «обрати файл ЕКГ для аналізу» та «обрати приклад ЕКГ для аналізу».

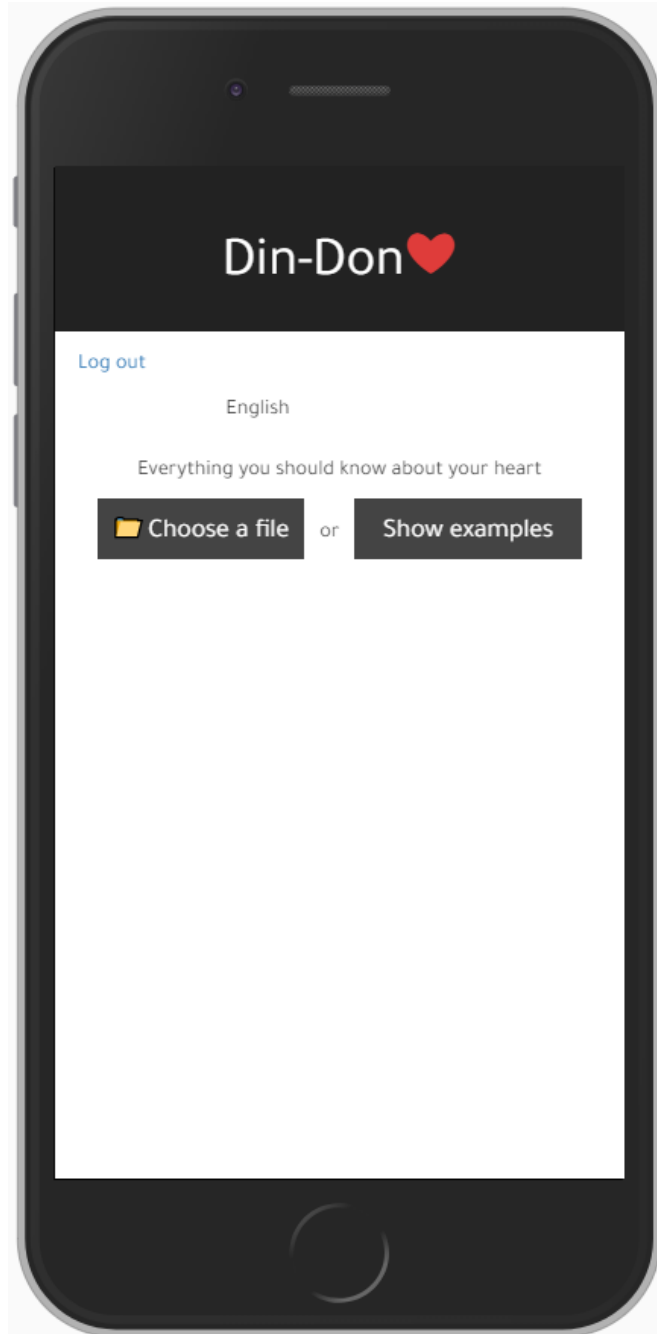


Рисунок 3.2 – Головна сторінка

Якщо користувач обирає функцію «обрати файл ЕКГ для аналізу», йому надається вікно завантаження знімку ЕКГ. Інтерфейс показано на рисунку 3.2

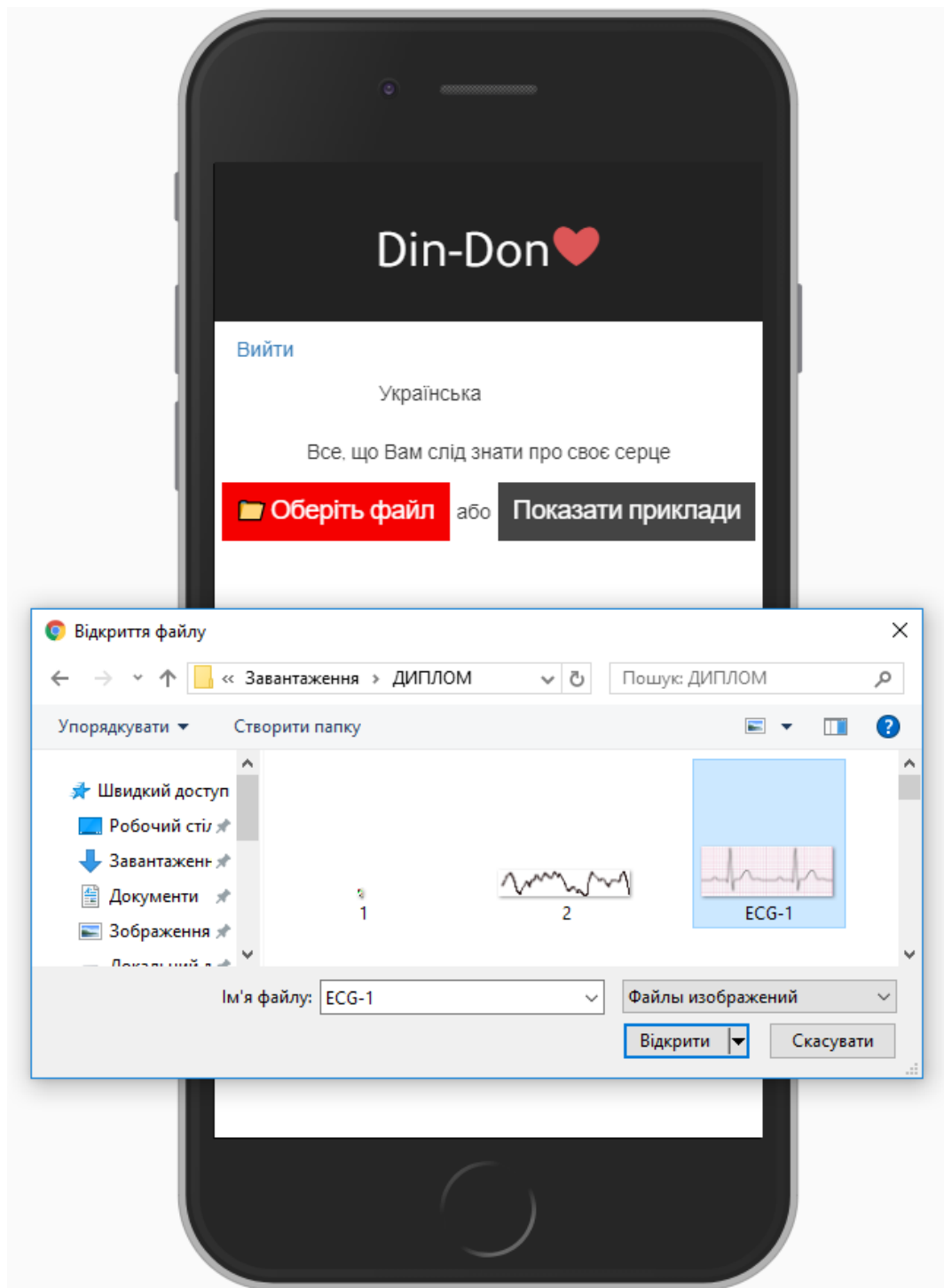


Рисунок 3.3 – Сторінка для завантаження знімку ЕКГ

Якщо у користувача на пристрої відсутні знімки ЕКГ або він хоче протестувати роботу веб-застосунку, йому надається можливість зробити це на вже встановлених прикладах. Він може обрати одне з трьох встановлених зображень для запуску аналізу.

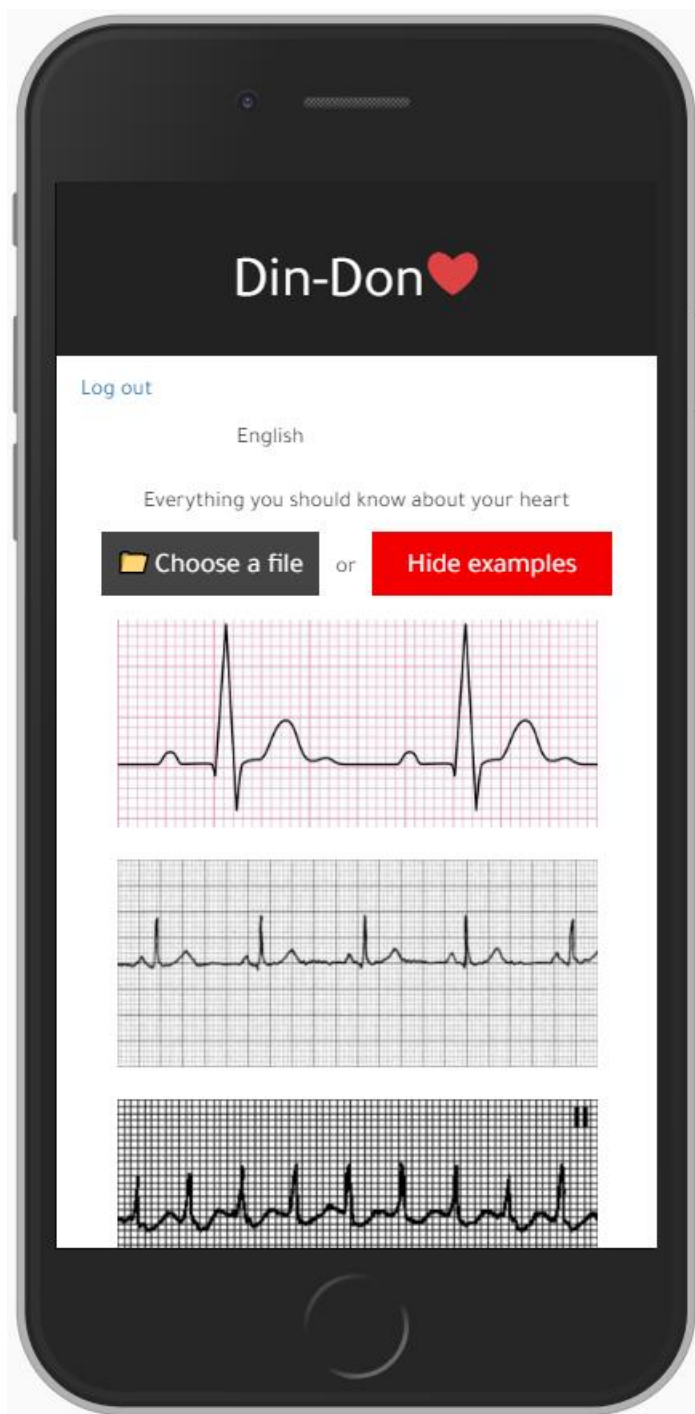


Рисунок 3.4 – Вибір тестового знімку ЕКГ для прикладу

Змн.	Арк.	№ докум.	Підпис	Дата

Після вибору знімку ЕКГ, не має значення, локального чи зі списку прикладів, запускається аналіз цього знімку. При завершенні аналізу результат виводиться на екран.

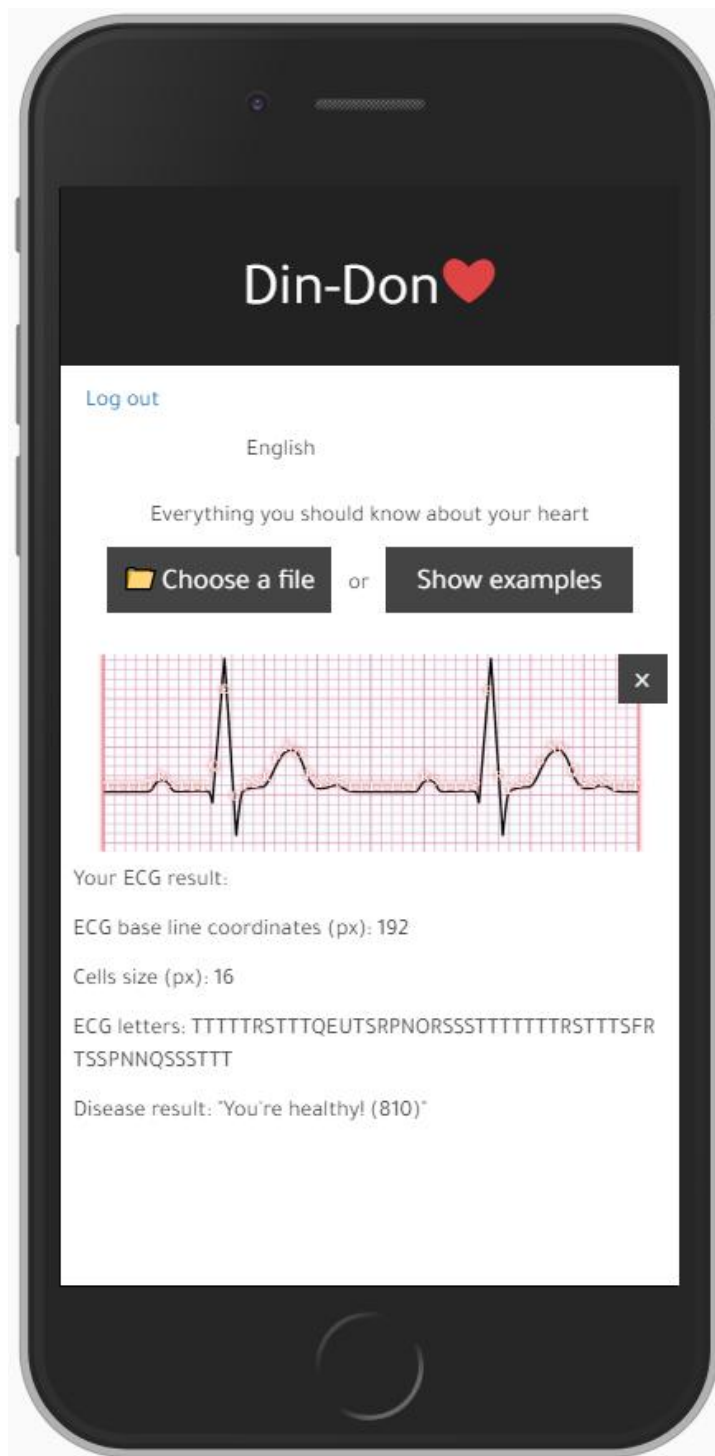


Рисунок 3.5 – Результат аналізу знімку ЕКГ

За замовчуванням, мова застосунку – англійська. Проте її можна змінити на свій вибір на одну з трьох: «англійська», «російська», «українська». Для цього треба обрати спеціальне вікно вибору мови.



Рисунок 3.6 – Вікно вибору мови

Змн.	Арк.	№ докум.	Підпис	Дата

Після вибору мови, наприклад, української, усі компоненти застосунку, що містять адаптований під локалізацію текст, також зміняться.

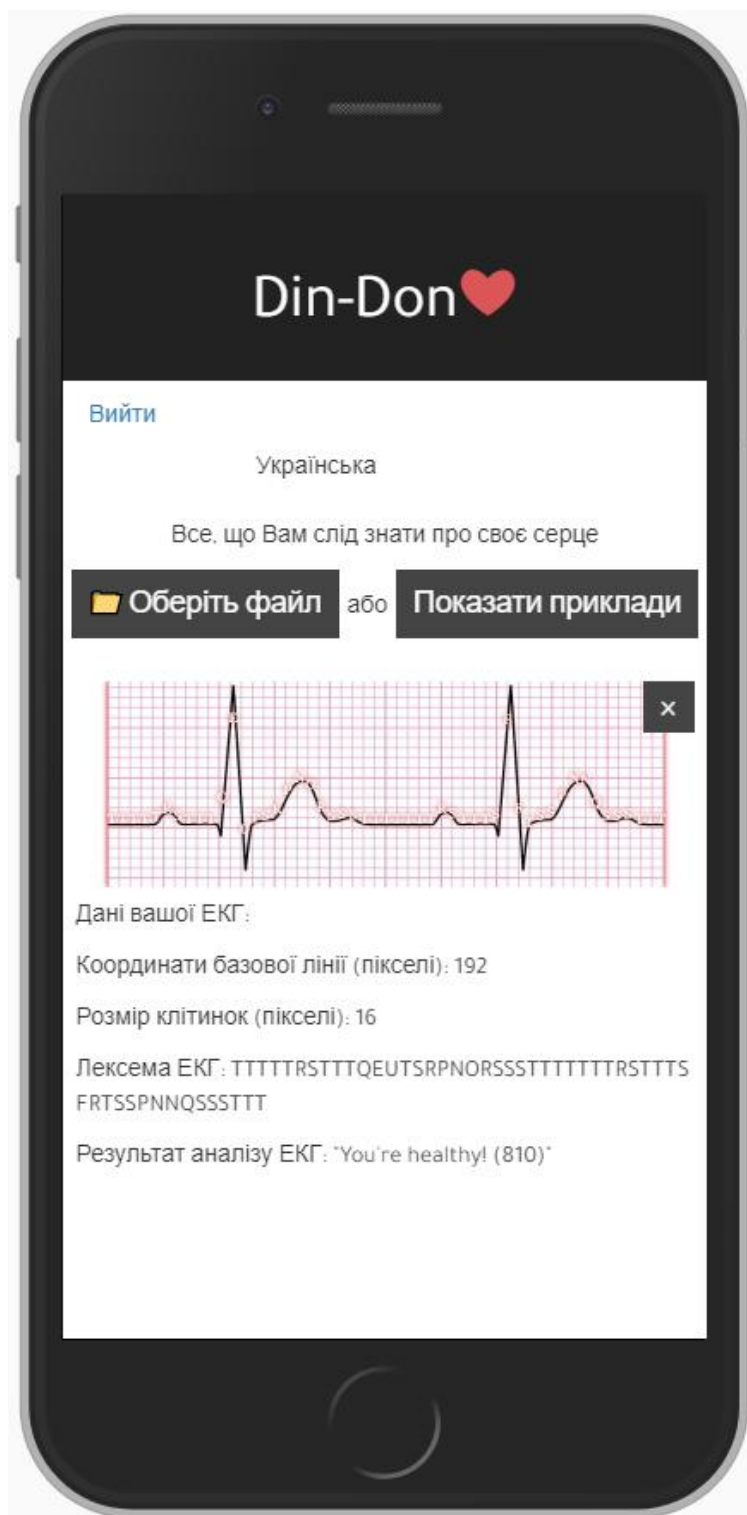


Рисунок 3.7 – Результат вибору української мови

Змн.	Арк.	№ докум.	Підпис	Дата

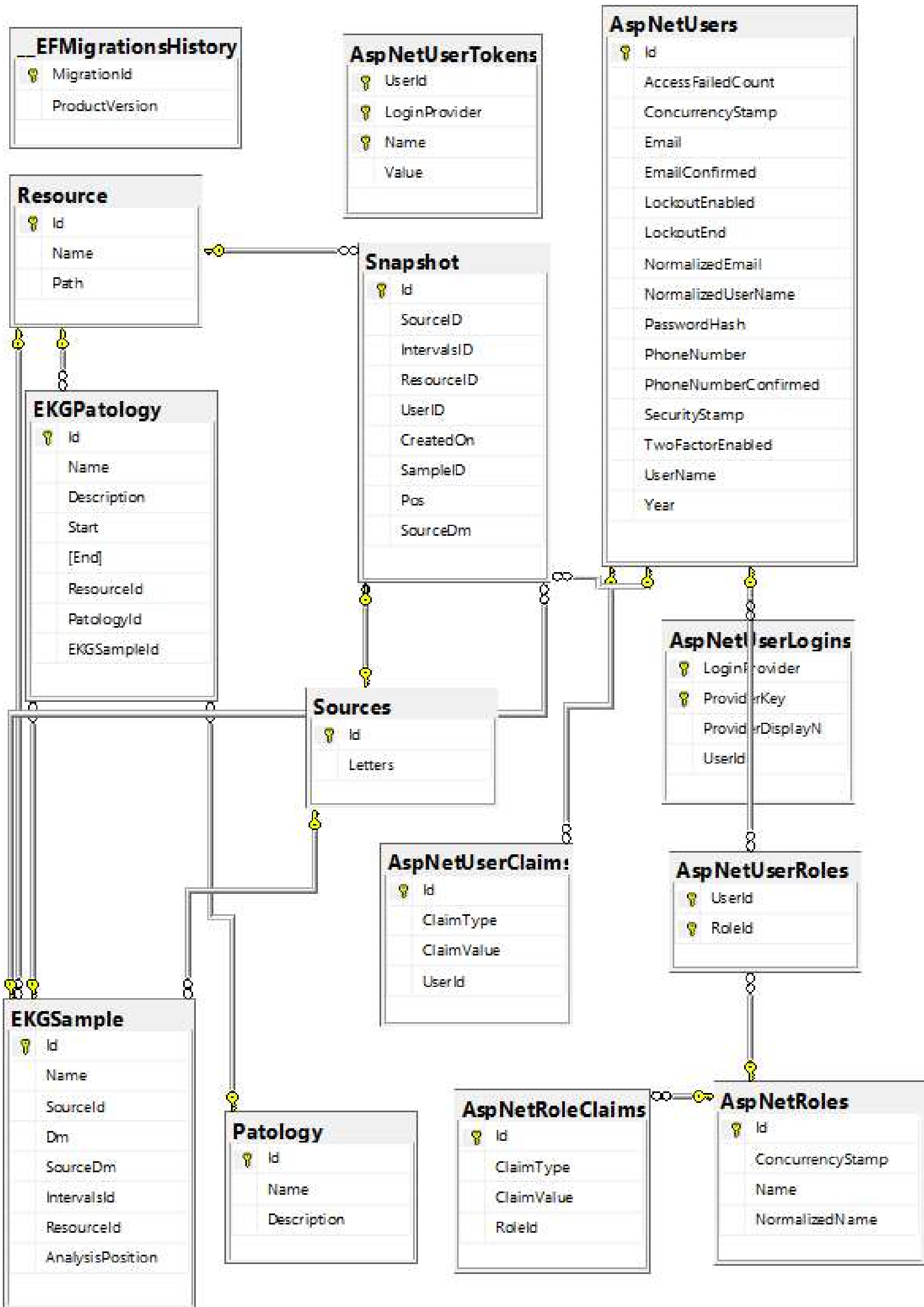
Якщо роль користувача – «Менеджер», то йому доступна функція завантаження прикладів хвороб. У спеціальній формі, яка надається йому після входу в систему, є поля опису хвороби.

Стисло про кожне з них:

- Mark – назва хвороби;
- Source – ЕКГ-лексема;
- Description – опис;
- Cells Size – орієнтований розмір клітинок.

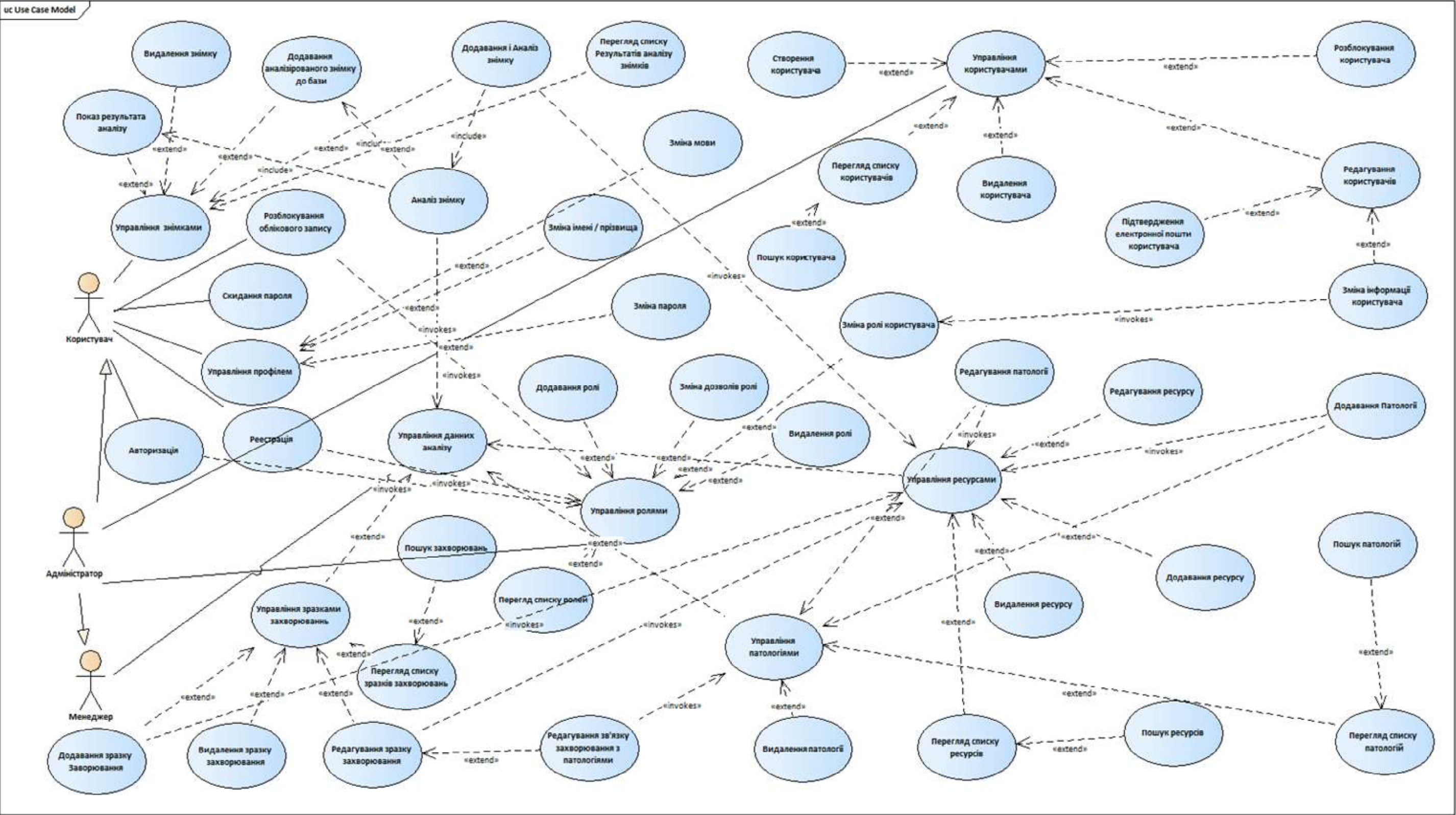
The screenshot shows a mobile application interface for adding a disease record. The screen is titled "Add" and shows fields for "Mark" (Arrhythmia), "Source" (a long alphanumeric string), "Description" (Heart rate pace is not constant through the time.), and "cells Size" (14). There is a "Create" button and a "Back to List" link. The footer shows "© 2018 - EKG".

Рисунок 3.8 – Форма опису хвороби

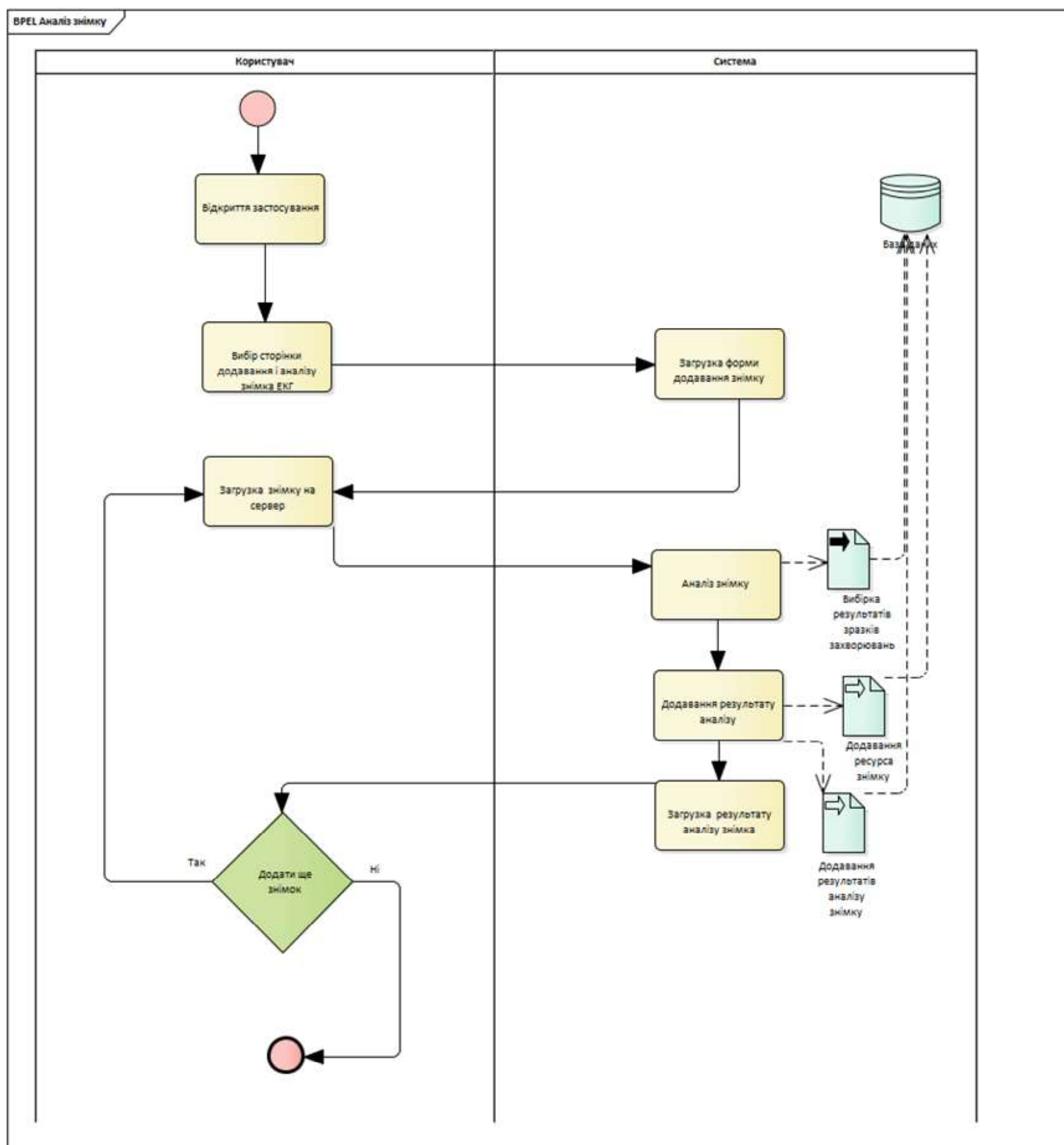


					КПІ-ІІІ-34111.045490-06-99.СБД			
					Схема бази даних	Лит.	Арк.	Аркуші
								1
						Аркуш		Аркуші
					Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини	КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-341		
Зм.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Сорочинський Р.В.						
Перев.		Баклан І.В.						
Т. Кон.								
Н. Кон.		Лішук К.І.						
Затв.		Баклан І.В.						

Підп. дата	
Інв. № дубл.	
Інв. № взаєм. підп.	
Підп. дата	
Інв. № підп.	

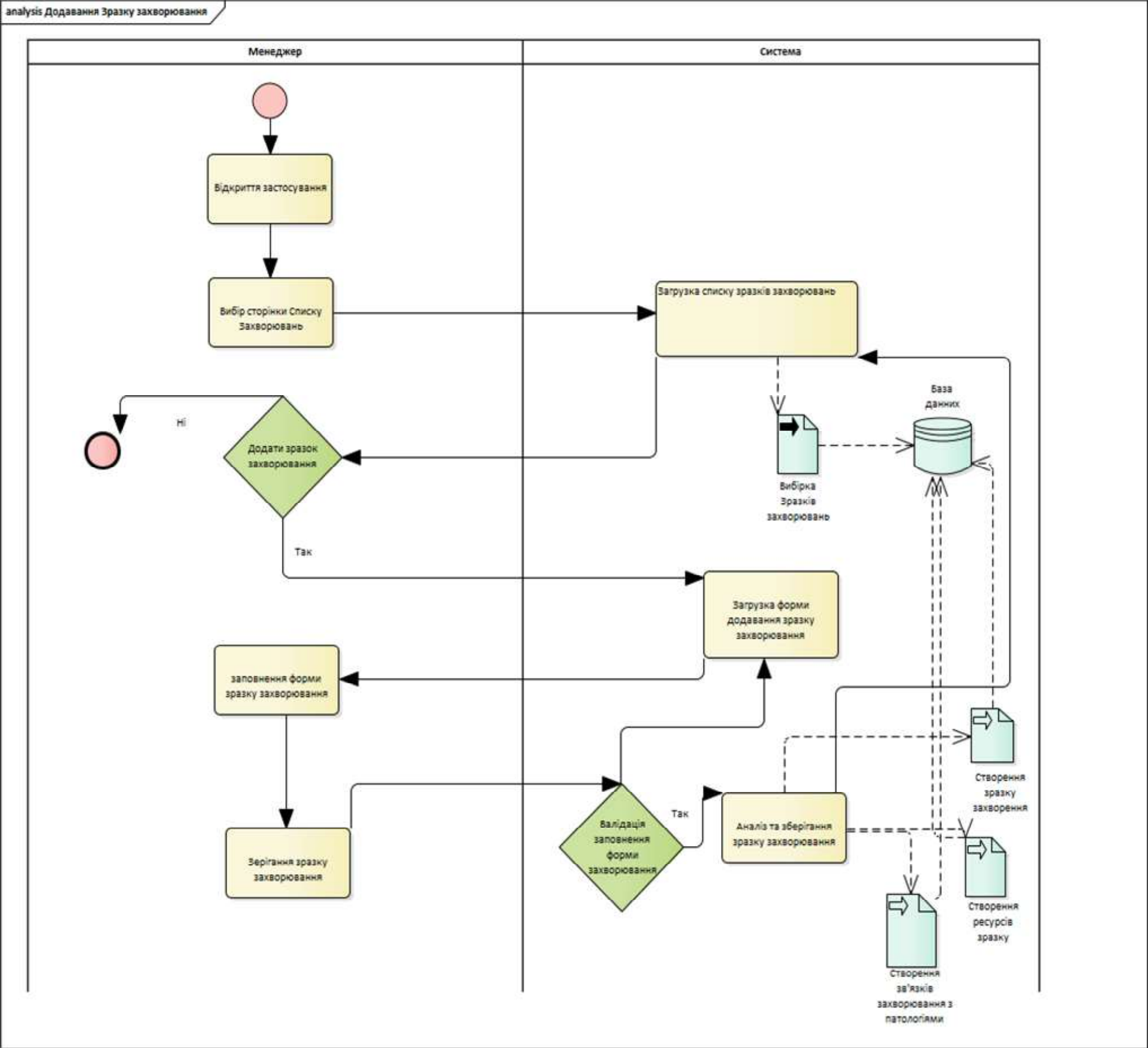


					КПІ-ІІІ-34111.045490-06-99.СС			
					Схема структурна варіантів використань	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив	Сорочинський Р.В.							
Перевірив	Баклан І.В.							
Т. кон.					Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини	Аркуш		Аркушів
Н. кон.	Лішук К.І.					КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІПз-41		
Затвердив	Баклан І.В.							



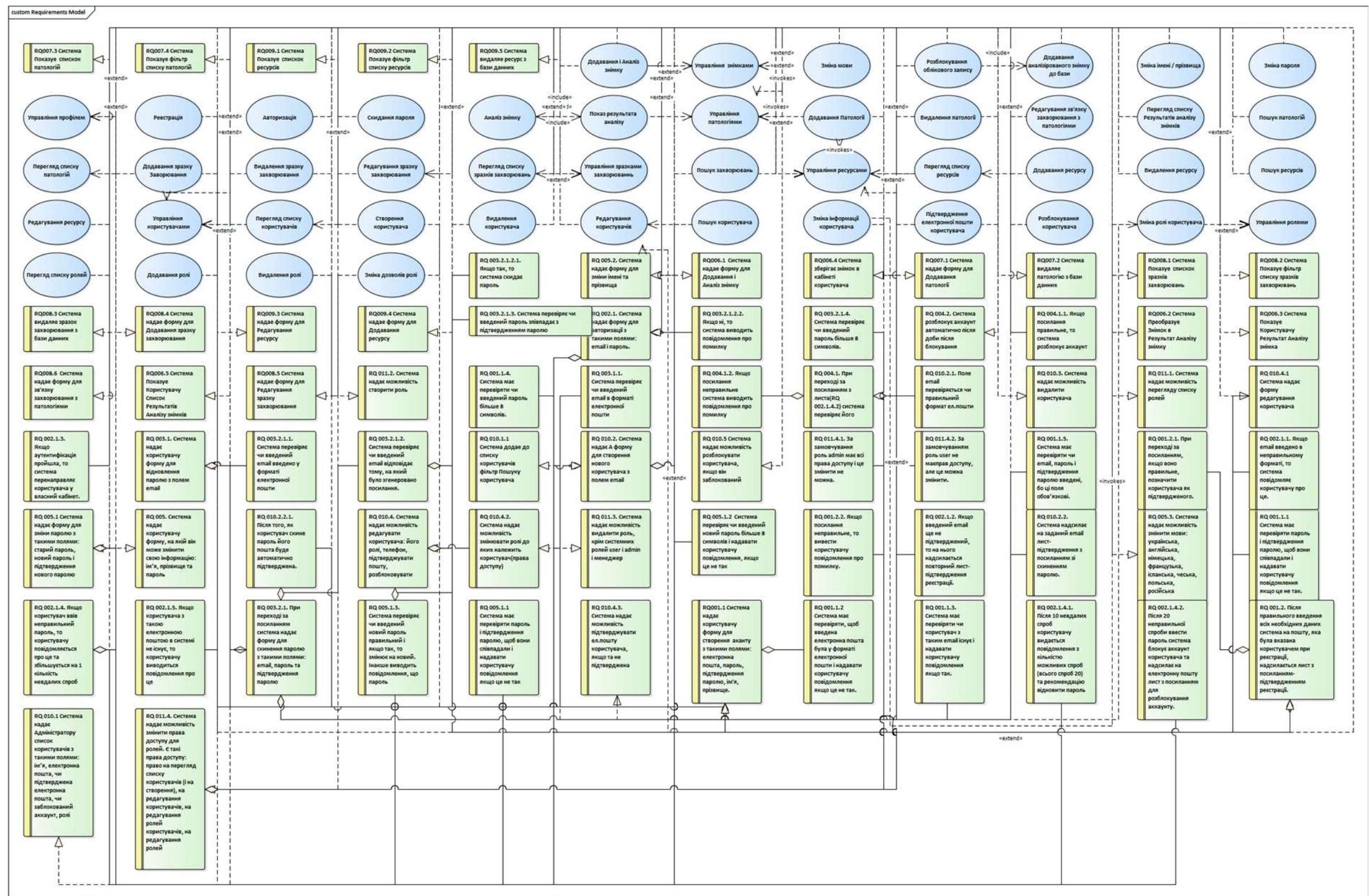
Інв. № підп.	Підп. дата	Інв. № взам. підп.	Інв. № дубл.	Підп. дата

					КПІ-ІІІ-34111.045490-06-99.СС					
					Схема структурна станів системи	Лит.			Арк.	Аркушів
										1
Зм.	Арк.	№ докум.	Підп.	Дата		Аркуш			Аркушів	
Розроб.		Сорочинський Р.В.				Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини				
Перев.		Баклан І.В.								
Т. Кон.					КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-341					
Н. Кон.		Ліщук К.І.								
Зам.		Баклан І.В.								

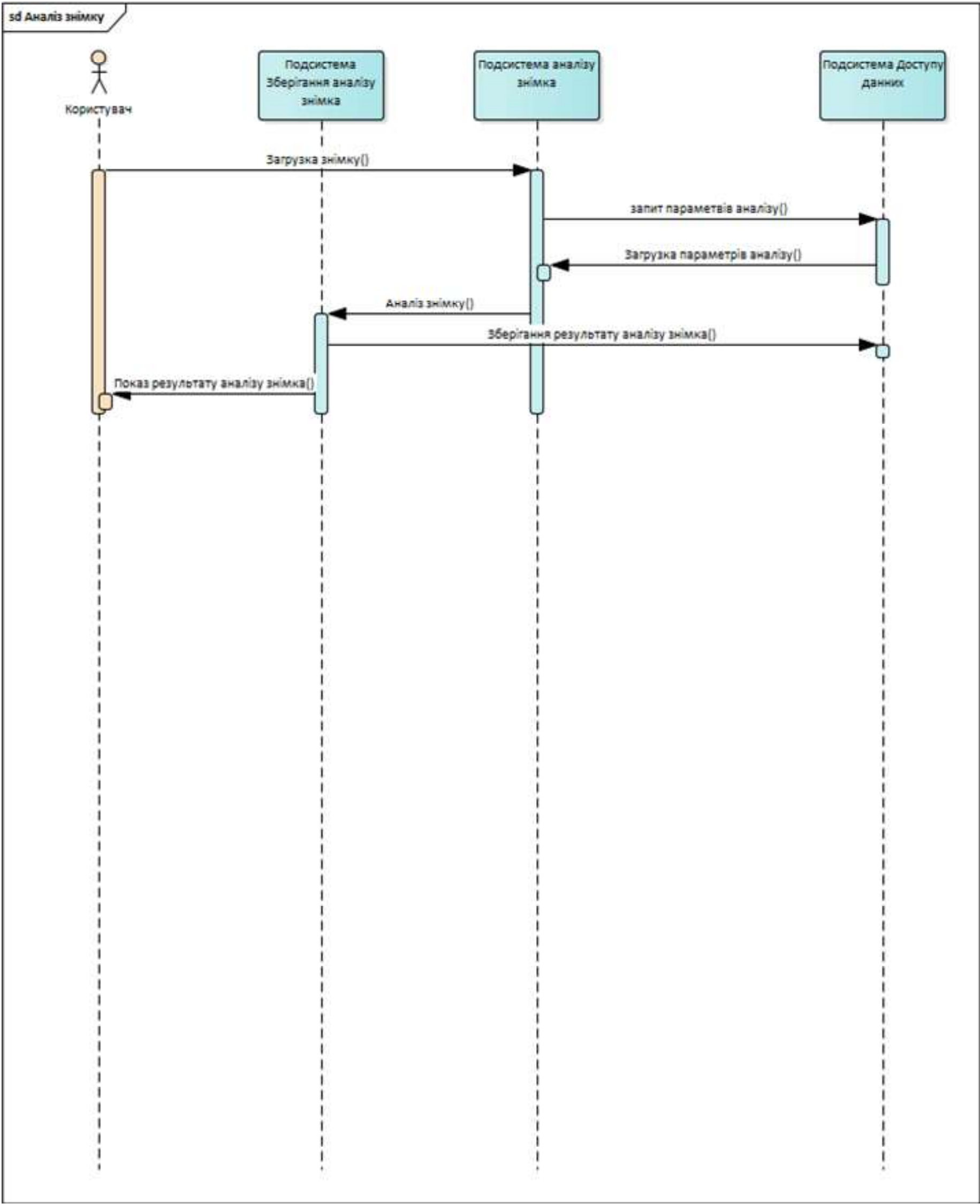


Інв. № підп.	Підп. дата	Інв. № взаєм. підп.	Інв. № дубл.	Підп. дата

					КПІ-ІІІ-34111.045490-06-99.СС					
					Схема структурна станів системи	Лит.		Арк.	Аркушів	
Зм.	Арк.	№ докум.	Підп.	Дата						1
Розроб.		Сорочинський Р.В.								
Перев.		Баклан І.В.								
Т. Кон.										
					Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини	Аркуш		Аркушів		
Н. Кон.		Ліщук К.І.				КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-41				
Затв.		Баклан І.В.								



					КПІ-ІП-34111.045490-06-99.СС					
					<p align="center">Схема структурна варіантів Використань з функціональними вимогами</p>	Літера			Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив	Сорочинський Р.В.									
Перевірів	Баклан І.В.									
Т. кон.										
					<p align="center">Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини</p>	Аркуш			Аркушів	
Н. кон.	Ліщук К.І.					<p align="center">КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІПз-41</p>				
Затвердив	Баклан І.В.									



					КПІ-ІІІ-з4111.045490-06-99.СС							
					Схема структурна послідовності	Лит.			Арк.		Аркушів	
											1	
						Аркуш			Аркушів			
						КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-41						
						Комплекс програм для віддаленого доступу до бази знань патернів аномалій роботи серця людини						
Зм.	Арк.	№ докум.	Підп.	Дата								
	Розроб.	Сорочинський Р.В.										
	Перев.	Баклан І.В.										
	Т. Кон.											
	Н. Кон.	Ліщук К.І.										
	Затв.	Баклан І.В.										

Підп. дата

Інв. № дубл.

Інв. № взаєм. підп.

Підп. дата

Інв. № підп.